TECHNISCHE UNIVERSITÄT MÜNCHEN
Institut für Informatik
Lehrstuhl für Netzarchitekturen und Netzdienste

# Optimization of Resilience in Virtual Networks

Işıl Burcu Barla Harter

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender:
Prüfer der Dissertation: 1. Univ.-Prof. Dr.-Ing. Georg Carle
2.

Die Dissertation wurde am ............... bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am ............... angenommen.

## Abstract

Business and consumer applications are increasingly based on communication networks and cloud solutions, which use private/public IT infrastructures. As a result, the load of the infrastructures and the dependency on these technologies are growing, and this trend is expected to continue in the future. Cloud services require reliable and high-quality end-to-end communications spanning network and IT domains. Moreover, the network functions are becoming virtualized and placed into the clouds, as in the case of Network Functions Virtualization, increasing the dependency of these two domains on each other. All these developments require flexible, reliable and efficient technological solutions and a cooperation of the network and IT domains. However, the current Internet architecture with its ossification problems and the current cloud solutions, where the IT and network domains are mainly operated by separate entities, cannot fulfill these needs. In this thesis, we propose end-to-end reliable network designs offering high performance for connectivity and cloud services using network virtualization with combined control of network and IT resources.

In a network virtualization environment, we expect new business roles to emerge. A Physical Infrastructure Provider (PIP) is the owner of the virtualized physical substrate, a Virtual Network Operator (VNO) operates virtual networks, which are set up using the virtual resources of the PIP domains, and a Service Provider (SP) requests connectivity and/or cloud services from the VNOs. In this architecture, there is the need and freedom to optimize the service routing inside the virtual networks and the mapping of the virtual resources onto the physical substrate to reach an overall optimization. This is an open issue up to now as the existing literature on the overlay networks, virtual private networks and network embedding provides only partial answers by assuming the mapping or the routing to be known a priori. We provide a solution to this problem by introducing novel end-to-end resilient virtual network design models, which enable the optimization of service routing and virtual network mapping simultaneously. We formulate our models as mixed-integer linear programmings and heuristic algorithms. We show using extensive simulations that the proposed models outperform prior approaches in terms of applicability, virtual network cost and complexity. Based on the mathematical models, we also show that our heuristic solutions are scalable and perform close to optimal.

The second open issue up to now is at which layer to provide resilience in a virtual network architecture. In our models and algorithms we consider three main alternatives, namely provisioning resilience in the virtual or physical layer or using a combination of both. We analyze the performance of these options in terms of virtual network setup cost, end-to-end service latency, network and IT resource requirements, virtual network complexity and failure coverage. The proposed models are compared using these metrics both qualitatively

and quantitatively to create a framework for the network providers and operators in their decision of resilience provisioning layer in the future.

Finally, we tackle the problem space of how different methods from traditional networks can be applied to network virtualization to increase the service quality and efficiency of these solutions. We focus our study on shared protection in virtual networks and Quality of Service (QoS) provisioning. We show that shared protection in virtual networks, enabling sharing of redundant virtual resources, lowers the cost of a virtual network for a VNO and increases the resource utilization efficiency for a PIP. Therefore, it creates a win-win situation for these business roles. Our results indicate that QoS provisioning in virtual networks is also a key requirement, which enables the VNOs and PIPs to offer quality guarantees for their services and enhances the efficiency of the network usage from a business perspective.

# Contents

# List of Figures

# List of Tables

# Part I

# Introduction and Background

# 1. Introduction

## 1.1 Network and IT Virtualization

The way people communicate and do business today is changing. Instead of calling people, we send messages or emails. We upload pictures and videos or post about what we are doing. These services are generally provided by servers located in large Data Centers (DCs). Before, many companies used to have various servers located in different locations, but now, they tend to outsource their IT services to cloud providers or locate them in private clouds within their company network. As a result, today's communication infrastructures consist not only of communication networks but also of storage and compute elements located in big DCs that constitute cloud infrastructures. Even the communication networks themselves will depend on clouds in the near future. Software Defined Networking (SDN), which is the separation of the control and data plane, and Network Virtualization technologies enable Network Functions Virtualization (NFV), where the basic idea is to locate the network elements' intelligence in the cloud and enable the use of standardized hardware within the communication networks.

Network virtualization is seen as a key enabler of future Internet and future networks. It decouples services from the underlying physical infrastructure. All the parts of the physical infrastructure, the network links, nodes and the servers, are virtualized. Each network resource or server can host multiple virtual resources simultaneously, which are rented to different service providers enabling a more efficient use of physical resources. An isolated complete virtual network contains these different virtual resource types, where isolation enables the operators to use their own layer-specific address space, protocol stack, routing and Quality of Service (QoS) definitions. Virtual networks mimic the whole functionality of a physical network and offer on top more flexibility in network design due to an overview of different physical network and cloud domains.

The concept of virtualization has its roots already in 1950s, where it started with time sharing, followed by virtual memory and finally by independence from hardware. There are different types of virtualization like server, application or desktop virtualization, however, in this thesis, we focus on the virtualization of the networks. The idea of decoupling services from the underlying infrastructure and creating an abstract network layer can be seen in technologies like Virtual Private Networks (VPNs) and overlay networks, which are widely used today. VPNs ensure only the isolation of the traffic on network links and overlay networks serve the coexistence and isolation of network nodes. The network virtualization concept that we are using in this thesis goes one step further and enables

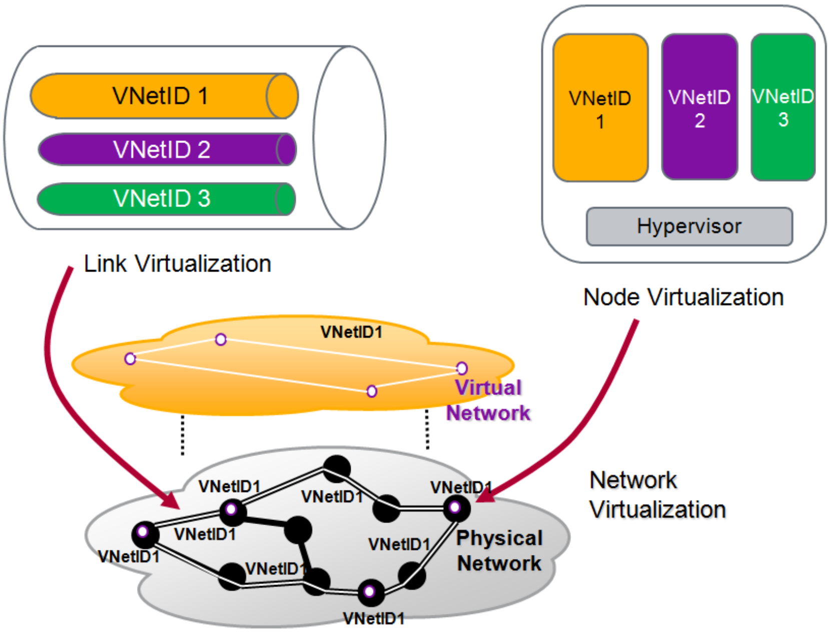


Figure 1.1: Network virtualization example: Physical links and nodes are partitioned into virtual resources. In the virtual layer, only a network consisting of its virtual resources is visible. Each virtual element and the entire virtual network are isolated from other virtual networks.

the virtualization of complete network slices, which can contain virtual links, nodes and e.g. Virtual Machines (VMs) on servers. An example about network virtualization is shown in Figure 1.1, which shows link and node virtualization. The physical resources are partitioned into isolated virtual resources. Each virtual network has access to and has the view of only its own virtual resources, which are labeled with its virtual network identity (VNetID). More details about the comparison of the existing virtualization technologies and the one used in the framework of this thesis are presented in Chapter 2.

Moreover, in a network virtualization environment, new business roles are expected to emerge. We use an architecture with three different business roles as shown in Figure 1.2. The Physical Infrastructure Providers (PIPs) are the owners of the physical infrastructure, which is virtualized and rented partially to the Virtual Network Operators (VNOs). The virtual resources rented by a VNO can be either only virtual link and node resources for deploying connectivity services, or a VNO can additionally also rent for example some VMs on servers to deploy end-to-end cloud services. Once the virtual network with the selected resources is established, the VNO can have full control on this virtual network. Finally, Service Providers (SPs) are the customers of the VNOs, which request connectivity or cloud services from them. More details about these business roles and the technological realization alternatives of such an architecture are given in Chapter 3.

Performance of cloud services is extremely critical for businesses. For example, Google reported 20% revenue loss due to a specific experiment that increased the time to display search results by as little as 500 milliseconds. Amazon reported a 1% sales decrease for an additional delay of as little as 100 milliseconds [1]. This performance requirement does not solely depend on the properties of the servers but also on the network connecting the users to these servers. Therefore, the connectivity to the DCs cannot be let to the best-effort service of the Internet. It is very important to have a complete high performance virtual network connecting the service source nodes and the cloud infrastructure sites. This is exactly the task of a VNO, which operates virtual networks that are optimized

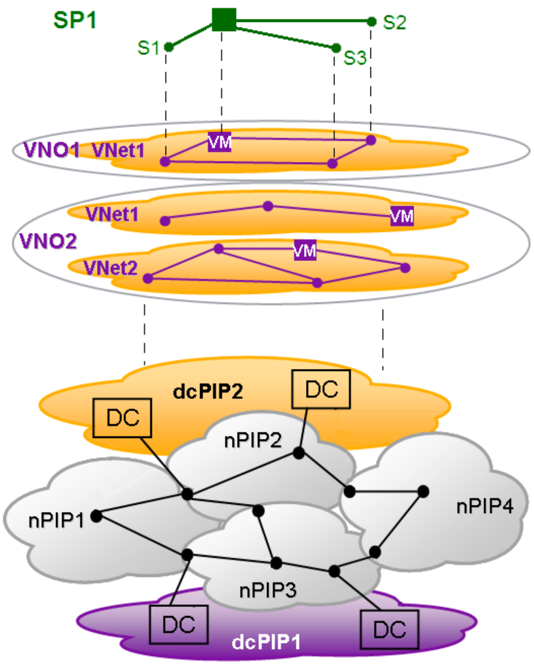


Figure 1.2: Virtual Network Environment (SP: Service Provider, S: Service source node, VNO: Virtual Network Operator, VNet: Virtual Network, VM: Virtual Machine, PIP: Physical Infrastructure Provider, nPIP: Network PIP, DC: Datacenter, dcPIP: Datacenter PIP)

in terms of cost and performance in an end-to-end fashion for cloud and connectivity services. To fulfill the requirements of the different service requests of the SPs efficiently, the VNOs should be in the position to design their virtual networks in an effective way using the advertised virtual resources of the PIPs. This requires a two step mapping of the service requests; through an optimal routing inside the virtual network down to the mapping of these virtual resources onto the physical substrate. In the literature about the overlay networks only the routing part is considered and the works about VPNs and virtual network embedding optimize solely the virtual network mapping, where in both cases the virtual network topology is assumed to be provided as an input. Therefore, these methods cannot be applied to the case of the virtual network design for a VNO and provide only sub-optimal solutions. Thus, the simultaneous optimization of virtual network mapping and service routing within this virtual network is an open research area up to now and is the main focus of this thesis. This solution is an enabler of flexible and cost-efficient future networks and cloud infrastructures with an end-to-end high performance and hence of new approaches like NFV.

In the Chapters 5 to 9, we introduce novel virtual network design models formulated as mathematical optimization problems and heuristics with different properties both for connectivity and cloud services. Via extensive simulations and analytical analyses, we evaluate the performance of the proposed models in each corresponding chapter. All of our models implement resilience mechanisms, which protect the virtual networks and the services against various forms of failures like single link or node failures, DC failures or even geographical failures. The reasons behind our special focus on resilience while designing virtual networks is elaborated in the next section.

## 1.2 Resilience

Resilience is the ability of a network or a system to maintain an acceptable level of service in case of hardware and software failures occurring within this network or system [2].

Resilience has been a key property of communication networks and will be increasingly important in the future networks due to ever increasing data rates and the dependency of our societies and businesses on the communication infrastructures. As mentioned in the previous section, IT and network domains are converging. Moreover, they become dependent on each other. Clouds require an adequate network connectivity and capacity for high performance access for the users, and networks need reliable and high performance cloud solutions for concepts like NFV.

While many businesses are outsourcing their IT services and switch to cloud services, reliability plays a crucial role in their decision for adopting these solutions. It is their primary concern according to a survey conducted with over 3700 companies worldwide [3]. Performance ranks third in the list of concerns and has about the same significance as the second one, namely security. Seeing the reliability concern at the highest rank is not surprising as service degradation and outages can be mission-critical or even fatal for businesses, and system outages are principally not avoidable. According to a research report [4] from end of 2013 based on 67 independent DCs located in the United States, 91% of the DCs have experienced an unplanned outage and the cost of an outage was on average over $7900 per minute in 2013. The service degradations and outages are not only due to the DC side but there can also be various network failures like fiber cuts, server or router failures, or even regional failures, which affect the communication network partially or entirely.

Therefore, it is crucial to offer an end-to-end resilient system incorporating both the network and the IT domains. However, today these two domains are mainly operated by separate entities, and hence, such an end-to-end optimization is mainly impossible. Network virtualization is a possible solution to this problem, which provides an overview of different and heterogeneous technology domains to a VNO, and hence enables a combined control of heterogeneous resources. Therefore, in this thesis we propose high reliability solutions in our virtual network designs.

This thesis addresses the topic of resilience in virtual networks from two perspectives. On the one hand, it proposes end-to-end reliable virtual network solutions for future connectivity and cloud services spanning different network and IT domains. On the other hand, in the framework of network virtualization, which is an enabler for future networks, this thesis answers the question of how to make the future networks resilient in a cost-efficient way. Regarding this second point, there are fundamental alternatives for resilience provisioning, namely resilience can be provided by a VNO or a PIP in their corresponding network levels, or a combination of these two concepts can be used. In other words, a VNO can either incorporate resilience into its virtual network design by using redundant virtual resources for the services or it can rent (at least partially) resilient resources from the PIPs and can have a simple virtual network design. In case of the latter, the PIPs are then responsible to ensure the reliability of the virtual resources by means like protection mapping of the virtual resources onto the physical substrate or e.g. by re-routing the traffic in the physical layer to another DC location in case of failure of the primary site. In the third option, namely when using a combination of these approaches, a VNO can incorporate a certain level of resilience into its virtual network design and can delegate certain protection tasks to the PIPs.

In conclusion, network virtualization is a key enabler for future networks and being able to design end-to-end resilient, cost-efficient and high performance virtual networks is crucial for the future network operators. In this thesis, this problem is converted into a mathematical modeling problem and novel virtual network design solutions are proposed incorporating various resilience mechanisms like shared protection as well as QoS guarantees. We also introduce scalable heuristic algorithms, which can design virtual network topologies with the given requirements close to optimum. Finally, we introduce novel resilience mechanisms for virtual networks, which provision resilience either in the virtual or physical layer or using a combination of both. We develop a simulator, which models the virtual network environment, and via extensive simulations we also evaluate the performance of all the proposed models. Our results form a framework for the decision on the resilience provisioning layer that shows the benefits and drawbacks of each alternative to the operators in their future network design. As mentioned before we also investigate the application of shared protection on virtual networks and provide solutions for having QoS-aware virtual networks. The complete list of the research questions that are addressed by this thesis is provided in the next section.

## 1.3 Open Issues

In this thesis we address four groups of research questions, which are presented in the following. In a network virtualization environment, the first essential problem is how a cost-efficient virtual network topology can be designed. The VNOs serve in general multiple SPs, and therefore, they want to reuse the rented virtual resources for different services in order to optimize the virtual network cost. At the same time, the services come with certain requirements concerning the routing within the virtual network and implicitly the mapping of the virtual resources on the physical infrastructure. Moreover, the VNOs typically do not have a total access on the physical topology of the PIPs but the PIPs advertise certain available resources to the VNOs, which can have different properties and hence different prices. Therefore, it is not trivial for a VNO to design a cost-efficient virtual network, which satisfies all service requirements.

In the literature, there is so far no direct solution to this problem. The literature about VPNs and virtual network embedding assume the virtual network topology to be given, which is either not possible for the case of a VNO or it would mean taking away the design freedom from the virtual layer by reducing the problem to a one-to-one mapping of the services on the virtual links. In case of the overlay networks, the virtual network topology is also given and mapped onto the physical topology and the only freedom is in routing the services within this topology. In conclusion, currently there is no work in the literature, which enables simultaneous optimization of the selection and mapping of the virtual resources and the routing of the services in the virtual layer. In this thesis, we address this issue, which is formulated in the following under the first research question group Q1. We incorporate resilience into our designs as it is a key network property. More details on the resilience topic and how it is handled in this thesis is explained in the following.

**Q1: Resilient virtual network design**

**Q1.1:** Does the prior art provide answers to the resilient virtual network design problem? If not, where are the shortcomings?

**Q1.2:** How can the design of resilient virtual networks be performed at the VNO layer using the input coming from the infrastructure providers, PIPs, and their customers, SPs?

**Q1.3:** How can resilient virtual network design be extended to cover cloud resources in order to provide end-to-end resilience for cloud services?

**Q1.4:** How can resilient virtual networks be designed to serve end-to-end resilient cloud service requests?

**Q1.5:** To cope with the possible scalability problems of the virtual network design models, what kind of heuristics can be used for resilient virtual network design?

As discussed in the former section, resilience plays a crucial role in today's networks and will keep and even increase its importance in the future. Therefore, it is very important to incorporate resilience in the virtual network design both for connectivity and cloud services. This is handled in research questions Q1.2-Q1.4. Moreover, the resilience issue raises the second most important question handled in this thesis, namely how to decide on the resilience provisioning layer. This is a design-time question a VNO needs to answer in order to decide if it should rent already resilient virtual resources from the PIPs or if it should incorporate resilience into its virtual network design, which causes the usage of an increased number but cheaper resources. Therefore, this thesis is dealing with the question of if it is better to provide resilience in the virtual layer or in the physical layer in terms virtual network setup cost, as well as resource utilization, service latency and virtual network complexity. Another option for resilience is to use a combination of the aforementioned alternatives, namely the hybrid methods. We evaluate what kind of advantages and disadvantages can be observed using these different alternatives. Finally, we also look at the trade-off between the resilience level and its cost to an operator and suggest a feasible level for the future operators. The research questions within this area that we answer in this thesis are listed in the following under the group Q2.

**Q2: Comparison of resilience provisioning at different layers**

**Q2.1:** What are the advantages and drawbacks of provisioning resilience in a certain layer in a virtual network architecture? In other words, does network virtualization offer any advantages in terms of resilience compared with traditional resilience provisioning?

**Q2.2:** Does virtual layer resilience bring any benefits in terms of virtual network setup cost, service latency, physical resource utilization and complexity?

**Q2.3:** What kind of failures can occur in a virtual network architecture and by which layer are they detectable and recoverable?

**Q2.4:** What is a feasible level of protection in terms of the failure coverage vs. cost trade-off in a virtual network?

Having resilience mechanisms increases the cost of a network due to redundant resource requirements. A method, which is common in physical networks and which offers a fast recovery with a reduced cost compared with dedicated protection, is shared protection. It allows the sharing of redundant resources by network flows, whose primary paths do not share any common failure risk, against which the system is protected. As for physical networks, the increased cost due to resilience is also a problem in virtual networks and can be solved by applying the shared protection concepts. However, due to the division of the network architecture into different business roles and the limited information sharing between these roles, this is unfortunately not straightforward. In the third research question group, Q3, we approach this problem by providing solutions for implementation of shared protection concepts in virtual networks and by evaluating their benefits with regards to dedicated protection schemes.

**Q3: Shared protection in virtual networks**

**Q3.1:** Shared protection is a widely used solution in physical networks offering reduced cost and fast recovery. How can it be applied to virtual networks?

**Q3.2:** What kind of architectural advances are necessary to enable the application of shared protection in virtual networks?

**Q3.3:** What are the design principles for allowing the usage of shared protection concepts in the framework of network virtualization?

**Q3.4:** How much gain does shared protection bring in virtual networks?

Besides resilience, the services usually have specific requirements like a certain end-to-end latency, which have to be considered in routing and virtual network design to reach customer satisfaction. In addition to the regular service classification, in a virtual network environment, multiple level classifications and their mapping need to be defined to model the relationship between the PIPs, VNOs and SPs. In this last part of the open issues, we investigate how QoS provisioning can be incorporated to the virtual network design, what are its benefits and at which layer it should be provisioned. These research questions are grouped in the following under Q4.

**Q4: QoS provisioning in virtual networks**

**Q4.1:** A good QoS provisioning is essential for customer satisfaction. How can that be done in the virtual network architecture?

**Q4.2:** At which layer is it better to provide QoS guarantees?

## 1.4 Outline and Contributions

In this section, we describe the contents of the following chapters briefly in the light of the answers they provide to the aforementioned research questions. Firstly, we present briefly the content of the first three chapters, which provide a thorough introduction to the proposed models by describing the background of this work, a literature survey, and the used framework and the tools. Afterwards, we introduce the Chapters 5 to 9, which are the core results chapters of this thesis. Finally, the conclusion chapters are briefly summarized.

Chapters 2 to 4 serve to provide the basic understanding of the background and the framework of the presented models. **Chapter 2** presents the background and the related work on the areas of virtualization technologies, resilience and optimization problems and heuristics. It clarifies the differences of the used network virtualization concepts with the existing technologies and shows how the existing literature in these areas relates to our work.

**Chapter 3** introduces the network virtualization architecture used throughout this thesis. It describes the technological details of this architecture as well as the participating business roles. Afterwards, it discusses how resilience can be provided in such an architecture highlighting the benefits and challenges this architecture causes for resilience provisioning. Finally, it presents an analytical analysis of the different layer resilience provisioning options, answering the research question Q2.1.

**Chapter 4** presents the Java Virtual Network Simulator, which has been developed within the framework of this thesis and is used in all the performance evaluations of the proposed models and algorithms.

**Chapter 5** describes in detail our resilient virtual network design models, which are formulated as Mixed-Integer Linear Programmings (MILPs). In this chapter, we tackle the open question of how to design resilient virtual networks, which cannot be answered up to now by the existing literature based on overlay networks, virtual private networks and virtual network embedding. All of them assume that the virtual network topology is known beforehand, which is not necessarily the case for virtual networks and are limited in their design either to only service routing in the virtual network or mapping of virtual resources. In this chapter, as our main contribution, we propose solutions enabling the optimization of virtual network mapping and service routing simultaneously. Thus, this chapter provides answers to the research questions Q1.1 and Q1.2. Moreover, it introduces the heuristic framework for resilient virtual design for cloud services, and hence, provides solutions to the research question Q1.5. Finally, it presents the performance evaluation of the proposed models via extensive simulations and compares the different layer resilience alternatives in terms of virtual network setup cost, service latency, physical resource utilization and complexity. Thus, it also provides answers to the research question Q2.2, which is one of the main research questions of this thesis, namely finding out the advantages and drawbacks of provisioning resilience at a certain layer in a virtual network architecture.

**Chapter 6** has a similar structure to its previous chapter, with the main difference that it addresses the resilient virtual network design problem for cloud services. Firstly, a thorough literature survey is presented with the special focus on cloud services providing answers to the research question Q1.1. Afterwards, the chapter goes step by step from the virtual network design models for connectivity services to the end-to-end optimized models for cloud services, which answers the research questions Q1.3 and Q1.4. The heuristic framework introduced in the former chapter is extended here to cover the provisioning of cloud services, providing further answers to the research question Q1.5. Finally, for the research question Q2.2, new solutions are provided, where the resilience layer decision evaluation is devoted to not only to the virtual or physical layer resilience provisioning but also hybrid approaches, which are a combination of the former two. This answer helps operators in the future to decide if they should leave resilience provisioning to the physical layer - as it is done traditionally - or if they should exploit a potential gain if they provision resilience themselves (at least partially) in the virtual layer.

**Chapter 7** deals with the application of the shared protection concepts to virtual networks. Shared protection is a widely used solution in physical networks offering reduced cost and fast recovery. In this chapter, we introduce a novel network architecture allowing the usage of this concept by virtual networks, allowing the sharing of redundant virtual resources. We show that shared protection can yield substantial network resource requirement and cost reductions in virtual networks, creating a win-win situation for the involved business roles. These contributions provide answers to the research questions Q3.1 to Q3.4. Moreover, the shared protection models are formulated both as MILPs and heuristic algorithms. We describe in detail the implementation of the HillClimber and kBest heuristics for resilient virtual network design with shared protection. We show that the proposed heuristics are scalable and perform close to optimal, which provides a matching answer to the research question Q1.5. Finally, we answer the research question Q2.2 showing the benefits and drawbacks of having resilience and shared protection at the two layers.

**Chapter 8** applies QoS provisioning to the design of virtual networks. A good QoS provisioning is essential for customer satisfaction in today's and future networks. For its application on virtual networks, we first define a classification framework for each relevant layer and show at which layer it is better to provide QoS guarantees. Thus, this chapter answers the research questions related to QoS provisioning, namely Q4.1 and Q4.2.

**Chapter 9** is the last core chapter and provides an extensive list of hardware and software failures, which might occur in a virtual network environment. Then, it analyzes these failure scenarios and discusses at which layer they are detectable and recoverable from. Extending the resilient virtual network design models for different protection levels, it shows what is a feasible level of protection in terms of the failure coverage vs. cost trade-off in a virtual network, and hence, answers the research questions Q2.3 and Q2.4. Moreover, it quantitatively compares the different layer-oriented resilience provisioning alternatives, providing answers to the research question Q2.2.

Finally, **Chapters 10** and **11** summarize the content of this thesis and provide an outlook. Chapter 10 especially focuses on the summary of the answers to the research question Q2.2, whose answer is divided into various chapters. Afterwards, Chapter 11 provides an overall summary and conclusion. To provide an overview of the respective chapters with the research questions they answer Figure 1.3 illustrates the mapping between them.

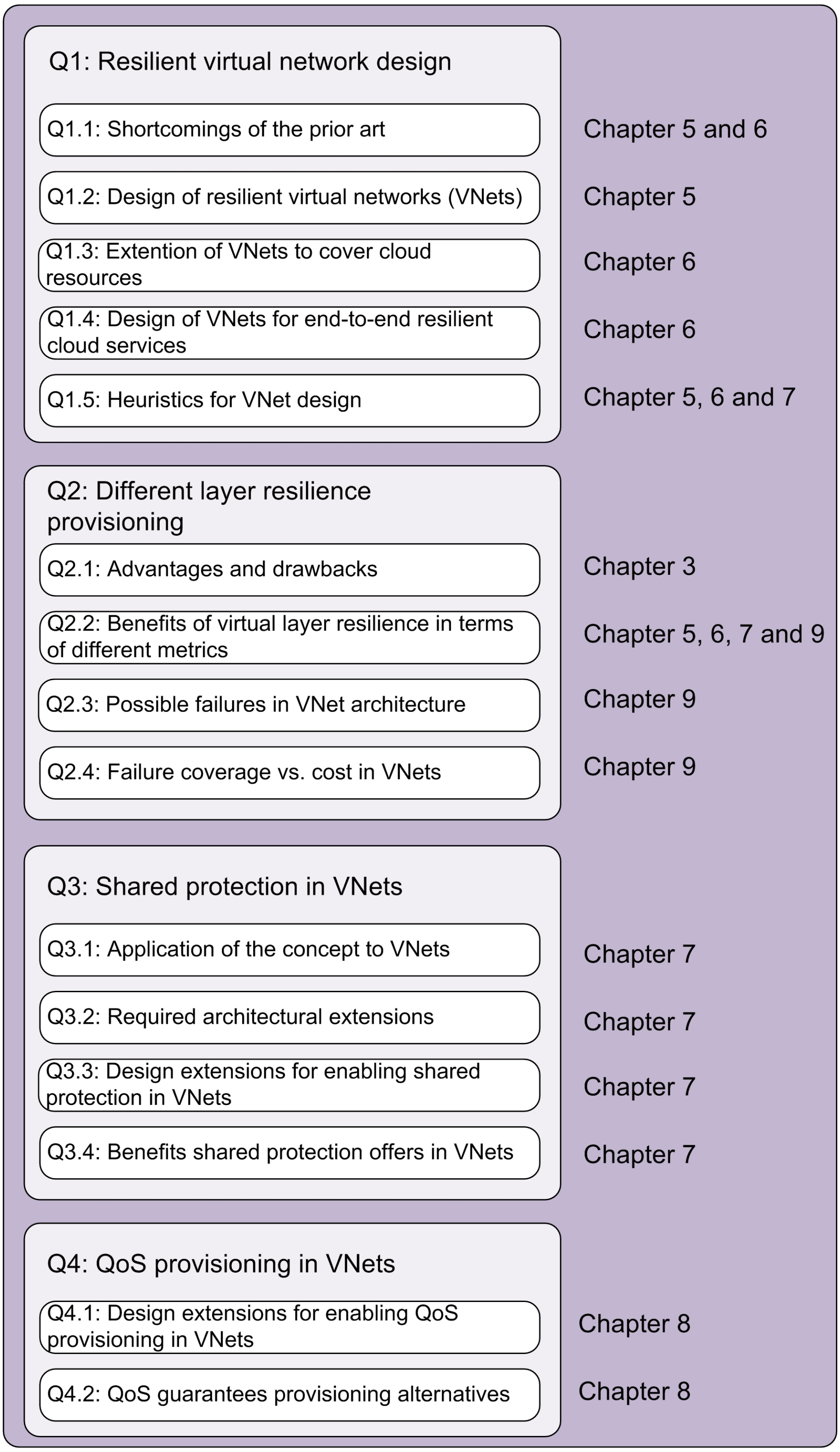

Q1: Resilient virtual network design
Q1.1: Shortcomings of the prior art
Chapter 5 and 6
Q1.2: Design of resilient virtual networks (VNets)
Chapter 5
Q1.3: Extention of VNets to cover cloud resources
Chapter 6
Q1.4: Design of VNets for end-to-end resilient cloud services
Chapter 6
Q1.5: Heuristics for VNet design
Chapter 5, 6 and 7
Q2: Different layer resilience provisioning
Q2.1: Advantages and drawbacks
Chapter 3
Q2.2: Benefits of virtual layer resilience in terms of different metrics
Chapter 5, 6, 7 and 9
Q2.3: Possible failures in VNet architecture
Chapter 9
Q2.4: Failure coverage vs. cost in VNets
Chapter 9
Q3: Shared protection in VNets
Q3.1: Application of the concept to VNets
Chapter 7
Q3.2: Required architectural extensions
Chapter 7
Q3.3: Design extensions for enabling shared protection in VNets
Chapter 7
Q3.4: Benefits shared protection offers in VNets
Chapter 7
Q4: QoS provisioning in VNets
Q4.1: Design extensions for enabling QoS provisioning in VNets
Chapter 8
Q4.2: QoS guarantees provisioning alternatives
Chapter 8

Figure 1.3: Research questions of the thesis. The chapters, providing answers to these research questions are listed across them.

## 1.5 Publications in the Context of this Thesis

This section presents the list of the publications in the context of this thesis in chronological order and the respective chapters they are related to.

- **Magazines and Journals**
  - I.B. Barla Harter, D.A. Schupke, M. Hoffmann, and G. Carle, ”Network Virtualization for Disaster Resilience of Cloud Services”, IEEE Communications Magazine, (Accepted). **(Chapters 3 and 9**)
  - I.B. Barla Harter, D.A. Schupke, M. Hoffmann, and G. Carle, ”Optimal Design of Resilient Virtual Networks”, Journal of Optical Communications and Networking (JOCN), Invited article (Accepted). **(Chapter 6)**
- **Conference Publications**
  - I.B. Barla, D.A. Schupke, and G. Carle, ”Analysis of resilience in virtual networks,” In 11th Würzburg Workshop on IP: Joint ITG and Euro-NF Workshop Visions of Future Generation Networks, 2011. **(Chapter 3)**
  - I.B. Barla, D.A. Schupke, and G. Carle, ”Virtual Network Simulator Architecture,” UKSim2012, Cambridge, UK, March 28-30, 2012. **(Chapter 4)**
  - I.B. Barla, D.A. Schupke, and G. Carle, ”Resilient Virtual Network Design for End-To-End Cloud Services,” NETWORKING 2012, Prague, Czech Republic, May 21-25, 2012. **(Chapter 5)**
  - I.B. Barla, D.A. Schupke, and G. Carle, ”Delay Performance of Resilient Cloud Services over Networks,” 10th IEEE International Symposium on Parallel and Distributed Processing with Applications (ISPA), International Workshop on on Cross-Stratum Optimization for Cloud Computing and Distributed Networked Applications, Madrid, July 10-13, 2012. **(Chapter 6)**
  - I.B. Barla, D.A. Schupke, M. Hoffmann and G. Carle, ”Optimal Design of Virtual Networks for Resilient Cloud Services,” International Conference on Design of Reliable Communication Networks (DRCN), Budapest, Hungary, March 4-7, 2013. **(Chapter 6)**
  - A. Basta, I.B. Barla, M. Hoffmann, G. Carle, and D.A. Schupke, ”Failure Coverage in Optimal Virtual Networks,” Optical Fiber Communication Conference and Exposition (OFC) and National Fiber Optic Engineers Conference (NFOEC), Anaheim, California, USA, March 17-21, 2013. **(Chapter 9)**
  - A. Basta, I.B. Barla, M. Hoffmann, and G. Carle, ”QoS-aware optimal resilient virtual networks,” IEEE International Conference on Communications (ICC), Budapest, Hungary, June 9-13, 2013. **(Chapter 8)**
  - I.B. Barla, K. Hoffmann, M. Hoffmann, D.A. Schupke, and G. Carle, ”Shared Protection in Virtual Networks,” Workshop on Clouds, Networks and Data Centers, IEEE International Conference on Communications (ICC), Budapest, Hungary, June 9-13, 2013. **(Chapter 7)**
  - M. Bui, B. Jaumard, I.B. Barla Harter, and C. Develder, ”Scalable algorithms for qos-aware virtual network mapping for cloud services,” 18th International Conference on Optical Networking Design and Modeling (ONDM), 2014. **(Chapter 8)**
  - I.B. Barla Harter, M. Hoffmann, D.A. Schupke, G. Carle, ”Scalable Resilient Virtual Network Design Algorithms for Cloud Services,” 6th International Workshop on Reliable Networks Design and Modeling (RNDM), Barcelona, November, 2014 (Accepted). **(Chapter 7)**

# 2. Background and Related Work

As introduced in Chapter 1, the focus of this thesis is resilience provisioning in virtual networks. This thesis deals with the questions of at which business role resilience should be provided and for each of the different layer alternatives how a cost or latency optimal virtual network design can be performed. Therefore, there are three main areas, on which this thesis is based, namely network virtualization, resilience, and optimization methods and heuristics. This chapter aims to provide the reader with the necessary background information and a summary of related work in all of these areas as a basis of the work presented in the next chapters. The literature surveys related to the specific topics handled in each chapter and the details of our contributions are presented in the corresponding chapters.

The aforementioned topics are introduced in the next three sections. Section 2.1 presents the background information about different virtualization technologies and discusses the main differences of the network virtualization concept used in this thesis with these technologies. Afterwards, in Section 2.2 the definition of the term resilience is provided as it is used in the framework of this thesis and resilience technologies existing in the network and data center areas are shortly described. Finally, Section 2.3 provides a short introduction to optimization models and heuristics and specifies the ones, which are used in this thesis.

## 2.1 Virtualization Technologies

Virtualization as a concept starts already in 1950s as time sharing [5] comes along, then virtual memory becomes common and afterwards it turns into the meaning of independence from hardware. In a general sense, virtualization can be defined as building abstractions and using them instead of real things. There are different types of virtualization used today like server and storage virtualization, which build the basis of the new cloud computing architectures [6], application virtualization, desktop virtualization [7], and finally network virtualization [8].

The term network virtualization itself has been used for different contexts in recent years and covers various technologies. Therefore, it is important to have a look at these different technologies and to specify, which technologies and concepts are used in the framework of this thesis.

### 2.1.1 Virtual Private Networks (VPNs)

In the traditional network architecture there are two options for service provisioning. In case of small networking needs, subscription can be seen as a usual choice. Its main advantage is that the subscriber does not have to deal with engineering, operating and managing the network and can also allow economic savings due to economics of scale effect. However, there are also circumstances when it may be preferable for a business to turn to private network ownership. A network owner, unlike the subscriber, has complete freedom in implementing the service and features it requires. It can also control and enhance the specific network capabilities needed for the services. Finally, when requiring a large network, buying the equipment can be also advantageous [9], however, it brings the cost and complexity associated with it. One way to overcome the disadvantages of network subscription and network ownership is to sell connectivity and bandwidth in the form of VPNs [10], which offer a middle ground between network subscription and network ownership. A VPN is mainly a collection of network resources taken from an underlying network [9]. VPNs can be used to create intranets, i.e. all sites in a VPN belong to the same enterprise, or extranets, i.e. various sites in a VPN can be owned by different enterprises [11].

The VPNs can be classified based on the protocol used in the VPN data plane as Layer 3, 2 or 1 VPNs [12].

Layer 3 VPN refers to the Layer 3 communication between a set of sites making use of a shared network infrastructure [13]. When realizing the VPN connections, tunneling mechanisms provide isolated communication between two end devices. Available tunneling mechanisms include (but are not limited to) according to [14]: Generic Routing Encapsulation (GRE) [15, 16], Internet Protocol (IP)-in-IP encapsulation [17, 18], Internet Protocol Security (IPsec) [19, 20], and Multi-Protocol Label Switching (MPLS) [21, 22].

Another option is building VPNs at Layer 2, which have the advantage of being agnostic to the higher-level protocols. However, the control plane traffic increases with the growth of Layer 2 VPN membership and with the number of supported VPN services [23]. The types of Layer 2 VPN are distinguished by the characteristics of the service that they offer to the customers. The Virtual Private Wire Service (VPWS) is point-to-point, and Virtual Private LAN Service (VPLS) is a Layer 2 service that emulates a Local Area Network (LAN) service across a Wide Area Network (WAN). There is also the possibility of an IP-only LAN-like Service (IPLS) [24].

Finally, the VPNs can be built also in Layer 1, which enable multiple virtual client-provisioned transport networks over a common Layer 1 core infrastructure. In large carrier networks it allows supporting multiple service networks over a shared transport network, where these service networks can be controlled and managed using Generalized Multi-Protocol Label Switching (GMPLS). In addition, Layer 1 VPN can support capabilities to offer innovative services to external clients [25].

### 2.1.2 Overlay Networks

An overlay network is an abstract network, which creates a virtual topology on top of a physical substrate. It consists of a collection of nodes implementing this network abstraction on top of the existing physical network [26]. These nodes are then connected via overlay layer links, which might have a single or multiple physical link mapping.

The authors of [8] state that the overlays are not geographically restricted and participation is completely voluntary. Due to the voluntarily nature of participation in lending the resources to the overlay network, significant expenditures are typically not involved. Moreover, the overlays are flexible and adaptable to changes and easily deployable in comparison to any other network. In the aforementioned work they also state, however, that Anderson et al. [27] show that standard overlays have their limitations to support radical architectural innovation in at least two ways. First, overlays have largely been seen as a way to deploy narrow fixes to specific problems of the Internet architecture like performance, availability, content distribution etc. without identifying how multiple overlays can interact in order to replace the underlying Internet architecture. Second, most overlays have been designed with the emphasis on deployment in the current Internet architecture, basically in the application layer on top of IP. Therefore, they are not capable of supporting radically different concepts.

### 2.1.3 Network Virtualization as a Whole

The term network virtualization usually recalls the current technologies like VPNs or overlay networks. As we have shortly introduced before, these technologies, however, offer only a partial virtualization. VPNs offer only traffic isolation and connectivity services in a virtual layer. Overlay network architectures are based on node virtualization. Even in testbeds like PlanetLab [28], which are designed for supporting innovative ideas, only node virtualization is realized and link isolation is missing.

The usage of network virtualization in this thesis denotes however that both link and node virtualization is in place. Thus, we assume the virtualization of a whole network rather than only link or node virtualization, which gives it the name Network Virtualization as a Whole. This allows to the operator of this virtual network to have full administrative control on it. Additionally, the operator is also in the position to fully customize its virtual network according to the running services within the virtual network.

This type of virtualization has been also the focus of numerous research efforts. These efforts mainly focus on creating testbeds, where researchers can have their customized virtual network slices, supporting development and evaluation of novel future network technologies. The Great Plains Environment for Network Innovation (GpENI) network virtualization architecture [29, 30], which is part of the Global Environment for Network Innovations (GENI) program in US [31], allows researchers to select their preferred routing software and desired routing protocols turning their customized network topology into a testbed network. Researchers can also inject networking events such as a link failure and a node failure or network traffic through the customized virtual network. Other examples of research efforts on network virtualization in other parts of the world are e.g. OneLab program in Europe [32] or Slice-Based Facility Architecture (SFA) [33] or AKARI [34] projects in Japan.

Cabo [35] goes one step further and extends network virtualization beyond its use for supporting shared experimental facilities. They claim that the support for virtual networks itself should be the basis of the architecture rather than this architecture serving as an evaluation platform. They allow coexistence of multiple virtual network architectures and divide the traditional Internet Service Provider (ISP) ecosystem into two business roles, one owning and providing the virtualized physical infrastructure and the second one providing services on top of this virtual network. This concept is further developed in works like 4WARD [36, 37], G-Lab [38] and GEYSERS [39] projects. In [36] the authors state that the current Internet architecture is already divided into three business roles, which are generally hidden within the same companies. In the current architecture, there are service providers, like Google, and ISPs like AT&T or Telekom, which own the physical substrate and can lease part of it, and also offer a connectivity service. To allow the concurrent existence of several, potentially service-tailored networks, a new level of indirection and abstraction can be introduced, namely these two roles within an ISP can be separated in the framework of network virtualization as the physical infrastructure provider and the virtual network operator. The latter uses the parts of the virtualized physical infrastructure, which it has rented, to offer e.g. connectivity services to the service providers. They additionally define a fourth role, which acts as a broker between these two roles in choosing and renting virtual resources. Our network virtualization model is based on these works, where more details are provided in Chapter 3.

Network virtualization together with SDN paves the way to future networks. Many ideas underlying the SDN technology, which advocates separating the data plane and the control plane, have evolved over the past 20 years (or more). The main difference compared with the past is that open interfaces such as OpenFlow [40] enable more innovation in controller platforms and applications [41]. SDN, by making network switches in the data plane simple packet forwarding devices and leaving a logically centralized software program to control the behavior of the entire network, introduces new possibilities for network management and configuration, which can improve certain aspects of current network management [42]. Moreover, like in [38] and [39], the IT side is also considered as a part of the future networks. IT (or cloud) resources are virtualized like the network resources and are part of the virtual networks, which can then offer end-to-end services to their customers.

The industry vision for future network technologies is that we will have very high network and service quality requirements in a future like 2020. It is expected that support of up to 1000 times capacity will be required in mobile networks. Latency levels are aimed to be in the millisecond range. The networks are required to be self-aware, energy-efficient and personalized for customer experience. Moreover, the network and cloud domains are merging and opening the way to new networking technologies like NFV [43]. All major vendors and operators in the telecommunication area are now working on this topic like Nokia [43], Ericsson [44], Alcatel-Lucent [45] and Huawei [46]. In the next section, more details on NFV is provided, which is a crucial building block of future networks.

#### 2.1.3.1 Network Functions Virtualization (NFV)

Communication networks today are populated with a large and increasing variety of specialized hardware appliances. This involves various problems. First, introducing new network services is troublesome as they usually require another variety, and maintaining all these equipments in terms of space and power is getting increasingly difficult. These complex hardware-based boxes mean high capital expenditures, high energy costs and complex operation for the network operators. Together with the facts that these hardware-based boxes' operation and maintenance is very costly and they cannot cope with the rapid innovation in technology and services, there is a need for new solutions [47].

NFV is seen as the solution of these problems, which is defined by European Telecommunications Standards Institute (ETSI) in [47] as a method to transform the way that network operators architect networks by evolving standard IT virtualization technology to consolidate many network equipment types onto industry standard high volume servers, switches and storage, which could be located in DCs, network nodes and in the end user premises, as illustrated in Figure 2.1. The network functions are decoupled from the specialized hardware and are implemented in software, which gives the freedom to move or instantiate these functions at necessary locations in the network without the necessity of buying or installing any new network equipment. This enable ease of network operation as well as reduction in hardware costs.

To summarize, with NFV the network functions are located into the cloud and this system needs to meet the high requirements as stated above, which include a high performance and very low latency levels. For this purpose, the communication between the cloud and the rest of the network plays a crucial role. To maintain the required performance and latency levels, the operators cannot rely on the best effort traffic like in the Internet. At this point, a careful design of virtual networks is necessary, which include and connect these resources and which offer end-to-end performance guarantees. This thesis is mainly dealing with this problem. Moreover, it is also essential to have a high reliability both

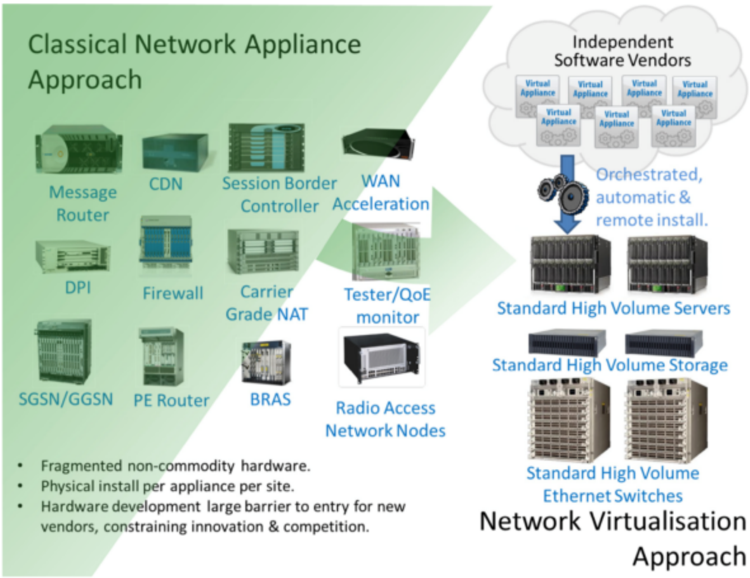

Classical Network Appliance Approach
Message Router
CDN
Session Border Controller
WAN Acceleration
DPI
Firewall
Carrier Grade NAT
Tester/QoE monitor
SGSN/GGSN
PE Router
BRAS
Radio Access Network Nodes
Fragmented non-commodity hardware.
Physical install per appliance per site.
Hardware development large barrier to entry for new vendors, constraining innovation & competition.
Independent Software Vendors
Virtual Appliance
Orchestrated, automatic & remote install.
Standard High Volume Servers
Standard High Volume Storage
Standard High Volume Ethernet Switches
Network Virtualisation Approach

Figure 2.1: Vision for Network Functions Virtualization [47]

on the cloud and the network side, since with the network functions placed in the cloud an outage of the connectivity can impact the whole network. Therefore, we include both network and cloud resilience in our virtual network models and optimize them usually in terms of virtual network cost, which is one of the main drivers for the operators as in the very case of NFV.

## 2.2 Resilience

The term resilience is used in various areas in slightly different meanings. Therefore, before going into more details of the resilience mechanisms, we first want to define our usage of this term. Resilience is the ability of a network or a system to maintain an acceptable level of service in the presence of hardware and software failures occurring inside this network or system.

As discussed before, the focus of this thesis is on the reliability of end-to-end connectivity and cloud services, which are provided in virtual networks. An example of a current network architecture, where cloud services are provided to the customers is shown in Figure 2.2. We limit our focus on the resilience of the transport network, which is shown as the ISP provider, and to the cloud side, which is referred as the content provider. Therefore, in this section, we introduce the current common practices in terms of resilience in the transport networks and on the cloud side in the two subsequent sections and highlight the technologies and mechanisms this thesis is using in the following chapters.

### 2.2.1 Resilience in Communication Networks

Transport networks, which are the focus of this thesis, enable local, regional and international transportation of voice and data traffic. The customer data is delivered by the transport network, which is a group of services and equipments that are responsible for reliable transportation, through switching and routing to the proper destination [49]. This destination might be large routers, some cloud installations and some other transport or access networks.

As the traffic and our society's dependency on communication networks is increasing day by day, having proper resilience mechanisms in place in communication networks is a must for customer satisfaction. Due to the high customer requirements, the providers

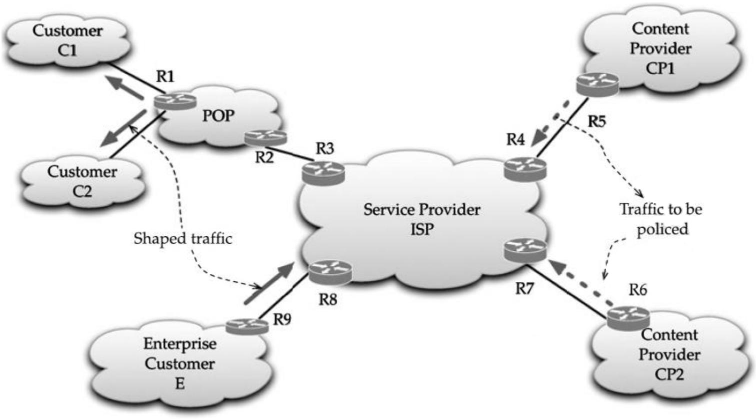


Figure 2.2: Example of an ISP network [48]

are committing and need to provide high availability rates like five 9s of resilience, which means that the connection has to be available 99.999% of the time, which corresponds to a network downtime of less than 5 minutes per year [50]. On the one hand, reaching such high availability rates is very demanding as failures due to fiber-cuts, which are the most common failure types [51], happen on average every four days [52]. On the other hand, a resilient communication infrastructure is crucial for businesses. According to a study conducted with companies from over seven different sectors [53], the revenue loss of a company due to network downtime could be in 2006 as high as 4 million pounds. Network outages can cause also productivity losses, which can be over 1.2 million pounds for a company. All in all, these facts give an insight for the importance and requirements of good resilience mechanisms.

Resilience mechanisms can be mainly divided into two groups, namely protection and restoration mechanisms, as shown in Figure 2.3. Protection mechanisms are based on pre-computation and reservation of the backup resources in the design time of the network. For each service connection working and protection paths are assigned. Working paths are used for data transmission under normal operation and protection paths are alternative paths, which are used in case of failure. Working and protection paths are diversely routed to prevent the loss of both paths in case of failure [50].

For restoration schemes, the spare redundant resources are determined online, and hence, there is no resource reservation in advance. Due to prevention of the spare capacity reservation during normal operation, they are more resource efficient compared to protection schemes. However, in contradiction with the protection schemes, they cannot guarantee the recovery of the services and are generally slower than the protection schemes in terms of recovery time [55]. Due to the fast recovery requirements, especially in optical networks, protection mechanisms are used for resilience against a single failure, which are the most

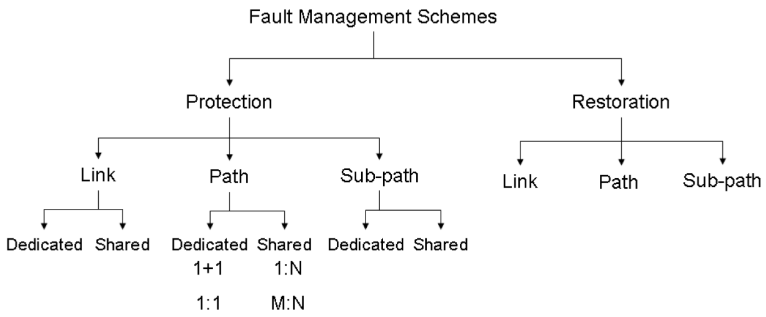
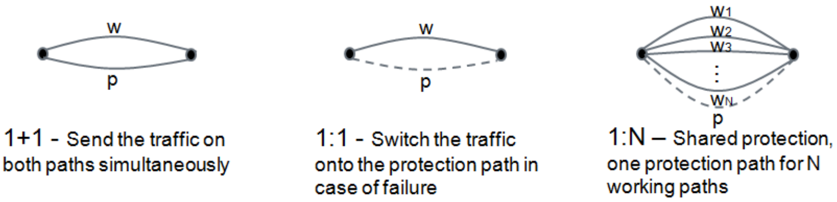


Figure 2.3: Classification of resilience schemes [54]



1+1 - Send the traffic on both paths simultaneously

1:1 - Switch the traffic onto the protection path in case of failure

1:N – Shared protection, one protection path for N working paths

Figure 2.4: Classification of protection mechanisms: $w$ stands for a working path and $p$ for a protection path

common failures. Since in our virtual design models we offer resilience mainly against single link or node failures, we focus on protection schemes in this thesis.

Protection schemes can be further divided into sub-groups according to the coverage of protection as link, segment and path protection [55, 56]. In link protection, a protection path is determined and reserved around each link when the connection is set up. If a link $l$ fails, the traffic is locally re-routed on the protection path of $l$. In case of segment protection, the protection resources are computed for a group of links within a connection. Finally, in path protection, the end-to-end working path of a service is protected by a disjoint protection path. The type of disjointness can be chosen according to the need of the services as for example link, node or sub-network disjointness, which means that the working and protection paths cannot share any link, node or sub-network, respectively. The advantage of link protection over path protection is that it is faster since the re-routing is done directly at the egress node of the failed link. However, path protection is more resource efficient, since it offers end-to-end protection.

These protection schemes are grouped as dedicated and shared protection. In dedicated protection, for each working link or path, a protection path is assigned. In case of failure, this assigned resource is used. 1+1 and 1:1 mechanisms are two types of dedicated protection as shown in Figure 2.4. In case of 1+1 the traffic is sent over both paths and the receiver can select the better one from the two signals, based primarily on the presence or absence of any of them. Therefore, this mechanism provides instantaneous recovery from failures. The difference of 1:1 is that the protection path is used in normal operation for low-priority traffic and the traffic is only switched to it in case of a failure. This is the protection scheme that we use in Chapters 5, 6 and 8.

In shared protection, the protection path is not assigned to a single working path but is shared by multiple working links or paths. Examples of shared path protection are 1:N and M:N protection schemes, where Figure 2.4 illustrates the 1:N protection. In this case, $N$ working paths are using a common protection path. To avoid the simultaneous failure of multiple working paths, these working paths need to be mutually disjoint. In case of failure at one, its traffic is then re-routed to the protection path. M:N is a generalized version of 1:N, where $N$ working paths are protected by $M$ protection paths. Therefore, if all working paths are mutually disjoint, this scheme provides resilience against $M$ independent failures. Or it can be used to protect against single failures, where $M$ working paths are allowed to fail at the same time. More details about shared protection is provided in Chapter 7.

### 2.2.2 Resilience in and between Data Centers

Similar to network resilience, DC resilience is the ability of an entire DC or of its parts to recover and maintain its normal operation in case of an equipment failure, power outage or other disruption.

The resilience level provided for a DC depends on the service requirements. Some services might withstand one-two hours downtime, whereas for business-critical services such outages might not be acceptable. In that case, the DC providers can prefer to have increased resilience as the cost of not preserving such services might be higher than the resilience investment in case of a long service outage. The article in [57] states that according to a survey of Information Technology and Intelligence Corp., while companies can't achieve zero downtime, one out of 10 companies said that they need more than 99.999% availability. Such availability requirements are understandable as according to [58], on average, businesses loose between $84,000 and $108,000 (US) for every hour of IT system downtime. Moreover, the article in [59] states that the businesses, which they surveyed, suffered per year 14 hours of IT outage, which according to half of these businesses damaged their reputation and according to 18% highly damaged it.

One way of providing DC resilience is having redundancy of the components and subsystems within the DC system. The idea is that the redundant synchronized element takes over the operation in case of failure of the primary element and continues to support the user services with this transition being ideally transparent to the user. Examples can be using redundant servers or power supplies, where a server would be then connected to primary and backup power supplies. Another example can be connecting each aggregate switch within the DC network to multiple core switches, which provides increased connectivity as well as enhanced resilience in case of a link or interface failure. The same could happen between top-of-rack and aggregate switches [60]. Finally, the redundancy concept can be used also for a complete DC by e.g. using two different utility providers, with the second one being the failover option.

The second option of DC resilience is using multiple DCs for reliability. The work in [1] points out that the in system DC resilience is realized by using multiple commodity servers, which by their own have high failure rates. However, using their combination and failover mechanisms a high level of intra-DC resilience can be achieved. They suggest to apply the same mechanism network wide, by lowering the individual resilience levels inside DCs by for example canceling the redundant uninterruptible power supplies or generators and using instead multiple DC locations. This option can be considered by a provider depending on the size, locations and requirements of its DCs.

The main advantage of using inter-DC resilience is that it also offers protection against geographical failures like natural disasters, where a complete DC or even the sub-network might fail due to power outages, flood, etc. Therefore, this topic has raised interest in the academic community where there are various works published about how such infrastructures can be designed efficiently [61, 62, 63]. The authors in [61] optimize the resource allocation in terms of communication cost and latency, for which they minimize the distance between the selected DCs. Besides latency, energy consumption also plays an important role in selecting the primary and protection DC sites, as power consumption and resilience are the two main drivers of the operational expenses of network and DC operators [63].

In case of inter-DC resilience, important metrics in choosing the operation mode are the requirements of the service in terms of recovery time and sensibility to interruption. There are three options used today, which are shared systems, hot standby and cold standby [64]. For all of these options, there is a primary and a Disaster Recovery (DR) site system and in the cold standby case there is additionally a backup site. In case of shared systems, both systems are used regularly and a continuous bidirectional synchronization takes place between the two systems. If there is a problem in one of them, the second one continues to serve the user, and therefore, there is no direct differentiation of the two systems. In case of hot standby, the DR site is unused during normal operation but is ready to run all the time. Its software, configuration and data level are continuously synchronized with the primary system. If there is a failure in the primary system, the user traffic is switched to the DR site via a quick failover. The difference of the cold standby with hot standby is that the DR site is available but is not up and running. Therefore, there is a need for a backup system. In case of a failure of the primary site, the DR site is restored from the backup system to an appropriate state and is initiated. The users need then to connect to the DR site, which might cause an interruption of especially delay-sensitive services.

In our studies, which are shown in the following chapters, we mainly focus on inter-DC resilience since we provide resilience against network and complete DC failures. Intra-DC resilience can be provided in addition by the DC operator, however, more details on this topic is out of the scope of this thesis.

## 2.3 Optimization Problems and Heuristics

In this thesis, as the means of resilient virtual network design, we are using optimization methods and heuristics. Therefore, this section aims to provide a brief overview of these two domains with having a special focus on the methods and algorithms used within this thesis.

### 2.3.1 Optimization Problems

Many network design problems are of type multi-commodity flow problems and therefore they either possess exact Linear Programming (LP) formulations or can be reasonably approximated with LP formulations [2]. These type of problems can be solved using the simplex method [65]. Although no polynomial time bound has been shown to hold for any version of the simplex method and the algorithm works in a time proportional to the number of optimization variables, it performs usually very well in practice [66, 2]. There are also important design problems like non-bifurcated routing, which involve non-linear features and are NP-complete. For such problems, LP formulations are too simplified and there is the need of using integer variables. Such problems are called MILP [2]. There are various types of optimization problems like linear, linear quadratic, non-linear etc. Due to our usage of only MILP in our models in this thesis, we will restrict our discussion on linear programming types and especially on MILPs and how it can be solved. A general formulation of linear optimization problems is given in the following:

$$\min \quad \mathbf{c}^T \mathbf{x} \tag{2.1}$$

$$\mathbf{A}\,\mathbf{x} \leq \mathbf{b} \tag{2.2}$$

$$\mathbf{l} \leq \mathbf{x} \leq \mathbf{u} \tag{2.3}$$

$$\mathbf{x} \in \mathbb{R}^n \tag{2.4}$$

$$x_i \in \mathbb{Z} \quad i \in I \tag{2.5}$$

The expression (2.1) can be in the form of minimization or maximization and is called the objective function of the optimization problem. The vector $\mathbf{x}$ of length $n$ holds the problem variables as specified with (2.4) and $\mathbf{c}$ is a vector with constant values. The inequality (2.2) shows the linear constraints of the problem, which the solution has to obey. $\mathbf{l}$ and $\mathbf{u}$ are lower and upper bounds on the variables $\mathbf{x}$, respectively, as denoted in (2.3). Finally, (2.5) determines the number of the integer variables, where the set of their indices is defined as $I$. The remaining variables, which are not required to be integer, are called continuous variables. The special case of integer variables, where the variable has to

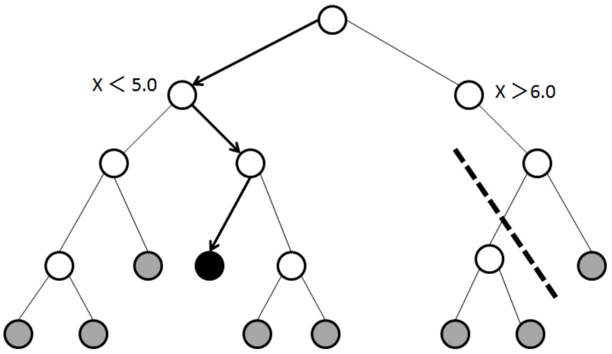


Figure 2.5: Branch and bound algorithm. Each node in the tree is a new MILP. The internal nodes of the tree, marked with white color, correspond to partial solutions. A bound can be used to eliminate infeasible or sub-optimal solutions as shown with the dashed line. Finally, the black node is found as the optimal point, where all leaf nodes, which are marked with gray color, can be solved or disposed.

take a value within the interval $[0, 1]$ are called binary variables, and are used for decision making purposes like e.g. if a demand is using a certain network link or not.

In the case that the set $I$ is empty, meaning that there are no integer variables, the problem becomes an LP. If all the variables are integer, i.e. if $|I| = n$, then the problem is called integer programming. A special case of it is when all integer variables are at the same time binary, where the problem takes the name binary programming. Finally, if there are both integer and continuous variables, it is called an MILP. MILP problems are generally solved using a linear-programming based branch-and-bound algorithm implemented in the well-known commercial solvers like CPLEX [67] or Gurobi [68].

### 2.3.1.1 Branch and Bound Algorithm

We have seen that the MILP problems are mostly NP-complete and as the real world problem sizes grow, these problems might not be able to be solved in a feasible time. Therefore, a good heuristic is needed, which eliminates parts of the search space where the absence of an optimum solution is known. Branch and bound is such a heuristic, which works based on the idea of successive partitioning of the search space [69].

For the description of the algorithm, we will mainly follow the practical explanation of the algorithm from [68]. The algorithm starts with the initial MILP, for which the direct solution is unknown. First, the integrality restrictions are removed, which turn the problem into an LP, and this LP is solved. This procedure is called LP relaxation. If the result actually satisfies all the integrality constraints of the initial MILP, the algorithm can stop at this point, however this is often not the case. The usual procedure is then to pick an integer variable, which has a fractional value in the LP relaxation. As an example, we assume an integer variable $x$, which has the value 5.7 in the LP relaxation. This value can be excluded by imposing the restrictions $x \leq 5.0$ and $x \geq 6.0$.

By inserting these restrictions we actually created two sub-MILPs having $x \leq 5.0$ and $x \geq 6.0$ imposed, respectively, as shown in Figure 2.5. In this case, the integer variable $x$ is called the branching variable, and the algorithm has branched on $x$ resulting in the two sub-MILPs. Now, the optimal solutions of these two sub-MILPs are calculated and the better one is taken, which is also a solution to the original MILP. Branching basically replaces the original MILP with two more stricter and hence simpler MILPs. The same procedure is then applied to these two sub-MILPs, where first the LP relaxation is applied and then if necessary branching variables are selected. By repetition of this procedure, the so called search tree is generated, where the sub-MILPs produced by the search procedure are the nodes of the tree and the initial MILP is the root node. The leaves are all the nodes from which the algorithm has not branched yet. At the end, if the algorithm reaches a point, where all leaf nodes can be solved or disposed, the original MILP is solved.

There is an additional logic, which is applied in processing of the nodes of the search tree. Assume that the goal of the MILP is minimizing the objective function. After solving a LP relaxation of some node, if all the integrality restrictions of the original MILP are satisfied, it means that a feasible solution to the original problem has been found. This has two important implications. First, there is no need to branch on this node anymore. It becomes a permanent leaf on the search tree, or in other words fathomed. Second, if the current value of the objective function with this feasible solution is better than the former saved value or if it is the first one available, this value is saved as the new incumbent, which means the best integer solution found at any point in the search.

There are two other possible reasons that can lead to a node being fathomed. First, if the LP relaxation at a node is infeasible due to an added restriction of that branch, the node doesn't have an integer feasible solution either. Second, if an optimal relaxation solution is found, whose objective value is higher than the current incumbent, this node cannot have a better integral solution. In both cases, this node is fathomed, i.e. the tree is pruned at that node.

There are two additional important values, which need to be introduced, to complete the description of branch and bound. First, assuming that the original MILP is a minimization problem, if we have an incumbent, it is a valid upper bound on the optimal solution of the MILP. This means that solutions with higher values than this bound can be directly discarded. Second, there is also a lower bound or best bound at any time in the search process, which is obtained by taking the minimum of the optimal objective values of all of the current leaf nodes. The difference between these upper and lower bounds is the optimization gap. When the gap goes to zero, the optimality of the end solution has been shown and the algorithm ends.

During the solution process the branch and bound algorithm can also make use of cutting planes, which are generally accepted to be the single most important contributor to the computational advances that have been made in integer programming over the last several years. The idea of cutting planes is to tighten the problem formulation by removing undesirable fractional solutions during the solution process. This is done via observation of the problem and including some additional constraints based on the allowed values the variables can take at that point. The solutions, which do not obey these, are cut off. This procedure, unlike branching, has the advantage that it doesn't have the undesirable side-effect of creating additional sub-problems [68].

For a more general description of the branch and bound algorithm and for other means of solving MILPs like branch and cut method, cutting-plane method and dynamic programming, the reader can refer to [2].

### 2.3.2 Short Overview on Heuristics

There are mainly two ways for solving optimization problems, either by the optimization methods as described in the preceding section or by means of heuristic algorithms. They are mainly used for solving problem instances, where the optimization methods fall short to find an optimal solution in a feasible time and/or using a feasible amount of resources. Heuristics are fast, however, they provide an approximate solution. Therefore, using heuristics brings a trade-off between optimality or precision and speed. A general method would be using the optimization methods for smaller instances of the problem, which then serve as an optimality benchmark for the heuristic algorithms, and solving the larger problem instances with heuristics, where the optimization methods do not scale. In the following, we present an overview on the different heuristic types based on [69].

A very basic heuristic is exhaustive search like enumerating, where the complete solution space is searched until the best global solution has been found. However, this kind of search method is not applicable for real world problems. Instead of exhaustively searching the entire solution space, another way is focusing on the neighborhood of a particular solution. In this method one applies a transformation to the current solution and evaluates the value of the new one. If the new solution is better, that one is kept and the procedure is repeated for its neighborhood. The key points in such an algorithm are defining the transformation function and determining the neighborhood size.

There are also some heuristics, which work with partial solutions and solve problems by constructing solutions one piece at a time. Greedy algorithms are one example of this group and they are popular due to their simplicity. It works by setting the values of the decision variables one by one by choosing the value with the best profit at each step. The major drawback of this approach is that taking optimum decisions at each separate step does not necessarily lead to a global optimum. Another example is the divide and conquer method, which tries to solve a complicated problem by dividing it into smaller simpler problems. The idea is solving each of these and then assembling the overall solution from them. This method is efficient if the cost of decomposing, solving and assembling the problem is less than the cost of solving directly the initial problem. Other examples are dynamic programming, $A^*$ algorithm and the branch and bound, which has been discussed in the previous chapter.

These algorithms can either find the global optimum but they might be expensive in doing that or they might get stuck in a local optimum. Therefore, there are some methods suggested like simulated annealing and tabu search, which overcome this problem. Simulated annealing has the basic idea that it accepts at some steps with a certain probability a worse solution with the hope of avoiding the local optima and reaching at the end the global optimum. Tabu search is a deterministic approach, where it uses a memory to save the information about the solutions, which have been examined recently. This allows to explore new areas of the search space, and it also helps to avoid local optima as the recently searched points become tabu - or forbidden - in making decisions about selecting the next solution. Finally, a newer type of algorithms, called genetic algorithms, mimic the process of natural selection by applying techniques inspired by inheritance, mutation, selection and cross-over.

Detailed descriptions of these heuristics are out of scope of this thesis and can be found in [69]. In the following, we focus on the two algorithms, which are used within this thesis for resilient virtual network design, and provide a short description of them.

#### 2.3.2.1 HillClimber Algorithm

In this thesis, we apply two heuristic algorithms in resilient virtual network design, the HillClimber and kBest algorithms. The former is described briefly in this section based on [70].

HillClimber algorithm with a random-restart is in the family of local search algorithms and can be described as an enhanced iterated Greedy algorithm, which aims to avoid the local optima. It is a simple algorithm with its inner loop continually moving in the direction of increasing or decreasing value when maximizing or minimizing the objective function, respectively. The algorithm does not maintain a search tree and only records the state and its evaluation. In each loop, the algorithm reaches a point at which no progress is being made. If this happens, an obvious thing to do is starting again from a different starting point.

One option to select the next starting point is doing it randomly. In that case, a series of the inner loops are run starting from random initial states until they make no discernible progress anymore. The best result from all these loops is saved at each iteration. The stopping condition of the algorithm can be either reaching a given maximum number of iterations or that the best value does not show any significant improvement. Instead of the randomness criterion, a different deterministic logic can be also used depending on the properties of the problem to be solved, which is also the case in our implementation. For the details of our implementation of the HillClimber algorithm for the virtual network design problem, please refer to Chapter 7.

#### 2.3.2.2 kBest Algorithm

The second algorithm we would like to introduce here is the kBest algorithm. It has been initially developed in the wireless networks area for the implementation of a maximum likelihood detector, which is the optimal receiver for multiple-input multiple-output channels [71]. In this thesis, we apply the general logic behind this algorithm to the resilient virtual network design problems.

The kBest algorithm uses breadth-first search instead of depth-first search. It starts with the initial problem and keeps the best $k$ solutions. At the next step, it continues from each of these points and creates for each of them new $k$ best solutions resulting in a total number of $k^2$ solutions at each step. From these solutions, the $k$ best ones are selected, and this procedure is repeated for all the decision points. The algorithm produces a tree structure with the final best $k$ leaves being the end of the paths leading from the root through the former decision points. The best solution path is then selected, which provides an overall result to the entire problem.

The implementation details and performance evaluation of the kBest algorithm for the virtual network design problem are provided in Chapter 7.

# Part II

# Framework and Tools

# 3. Framework: Network Virtualization and Resilience

This thesis is dealing with the problem of how and where resilience should be provided in a virtual network environment. The following chapters introduce various detailed models and algorithms and discuss their performance to answer these questions. Before going into the detail of these solutions, in this chapter, we introduce the framework we use throughout this thesis. First, our network virtualization model is presented. In a virtual network environment, we expect new business roles to emerge, which are introduced in detail in the next section with the tasks they realize and with their interaction with the other roles. We also discuss the technological alternatives in realizing this environment.

The second main part of this chapter is about resilience in the framework of network virtualization. We first discuss why resilience is a crucial issue and how network virtualization can be adopted to help with certain resilience problems. Afterwards, we present an analytical analysis in terms of resource utilization, service-level resilience and network setup and operation complexity of the different layer resilience alternatives.

The description of the virtualization model and the architectural details are based on our work in [72], and the analytical analysis presented in this chapter is an extended version of our work in [73].

## 3.1 Proposed Network Virtualization Model

Network virtualization is seen as a key enabler for future Internet and future networks. It offers more efficient resource utilization, flexibility and isolation of individual virtual networks. In this environment, new business roles are expected to emerge [36, 38], which realize different tasks. They interact with each other for the rental of virtual resources and setup of virtual networks. In our architecture, we define three main business roles, as shown in Figure 3.1. Note that these business roles can be combined into a single company or can be a separate business entity by their own. In the following, the properties of these business roles are introduced.

### 3.1.1 Physical Infrastructure Provider (PIP)

The Physical Infrastructure Provider (PIP) is the owner of the physical infrastructure, which can consist of fixed or mobile networks (Layer 1, 2 or 3) and IT resources like compute and storage, or any combination of them. It can monitor all of its physical and virtual resources and has the knowledge of the usage and location of its physical and virtual resources. The PIP optimizes the utilization of its network by allocating its virtual resources accordingly. Note that virtual resources can be also shifted from one physical resource to another one by the PIP if needed for overall optimization of the residing virtual networks or for shutting down a part of the network for energy efficiency and maintenance purposes. However, e.g. if resilience mechanisms are in place in the virtual layer, certain constraints about the physical disjointness of the virtual resources should be taken into account, which should be agreed on with the operator of the virtual network.

The physical infrastructure can be composed of multiple PIP domains. The choice of technology in the communication network is not limited; it can be Wavelength-Division Multiplexing (WDM), Ethernet, IP, etc. A PIP can fully control and monitor its resources, where it can use different management and/or control plane approaches like a GMPLS control plane or an SDN-based approach like OpenFlow (OF) [40], which introduces a standardized control interface and programmability to the switches by the use of flow-tables.

The PIPs are divided as Network PIPs (nPIPs) and Data Center PIPs (dcPIPs), depending on if they possess solely network resources and/or IT (or DC) resources, respectively. A dcPIP is expected to have its own DC network with various interconnected servers. It can also rent certain network resources from nPIPs to connect its DC resources. The interface between the DC and the WAN depends on the technologies used on both sides. The proposed models in this thesis are independent of the technological details of this interface, however, some examples are listed in the following to show how this interface can be implemented. For example, if MPLS is used in the DC, one can easily connect it to the GMPLS WAN with, e.g., hierarchical Label Switched Paths (LSPs) or LSP stitching. If OF is used in the DC, the OF controller can communicate with other OF controllers and with GMPLS. For the non-MPLS IP VPN and IP overlays not based on VPN like Virtual Extensible Local Area Network (VXLAN) in the DC, the connection can go over an Autonomous System Border Router (ASBR) and a datacenter Gateway (GW). In case of the former, there are different options like back-to-back Virtual Routing and Forwarding (VRF), External Border Gateway Protocol (EBGP) redistribution of labeled VPN-IP routes between neighboring Autonomous Systems (ASs) without and with multi-hop EBGP redistribution of labeled VPN-IP routes between source and destination ASs, listed in increasing scalability and decreasing security order [74]. For the latter, network overlay stitching can be applied using a DC-WAN GW performing, e.g., VRF termination or translation between the virtual network identities on the DC side and VPN labels on the WAN side [74].

### 3.1.2 Virtual Network Operator (VNO)

The resources of the PIPs are virtualized and advertised to the Virtual Network Operators (VNOs), where these resources can be virtual network links, nodes and e.g. virtual machines inside the servers. A VNO selects the resources it requires and requests the setup of a virtual network with these resources from the PIP(s). Once the virtual network is established, the VNO can have full control over it using its own control and management plane. A combined control of virtualized network and IT resources can be used enabling an end-to-end design and recovery for connectivity and cloud services, regardless of whether these resources belong to various PIPs or heterogeneous networks.

Interfaces, information sharing and pricing policy between a VNO and a PIP depend on their business models, and the contract between them [75]. In general, the aim of a VNO is to design cost-efficient virtual networks, while a PIP's aim is to maximize its profit by e.g. utilizing its resources in an utmost efficient way in order to be able to serve a maximum number of customers, the VNOs, with the same amount of resources.

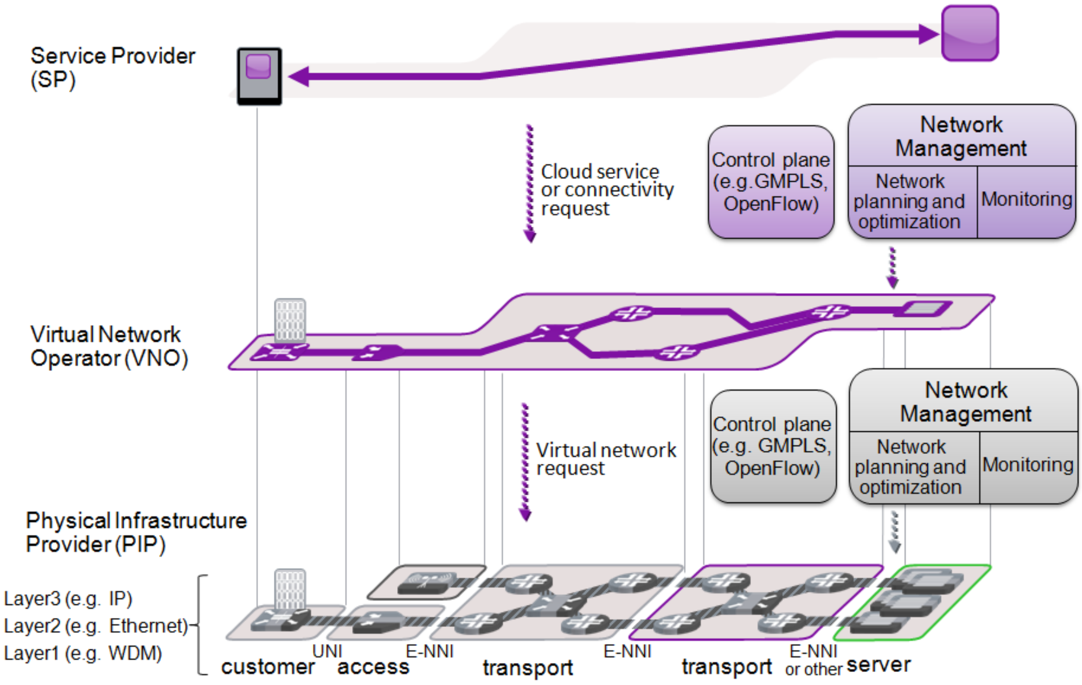

Service Provider (SP)
Cloud service or connectivity request
Control plane (e.g.GMPLS, OpenFlow)
Network Management
Network planning and optimization
Monitoring
Virtual Network Operator (VNO)
Virtual network request
Control plane (e.g. GMPLS, OpenFlow)
Network Management
Network planning and optimization
Monitoring
Physical Infrastructure Provider (PIP)
Layer3 (e.g. IP)
Layer2 (e.g. Ethernet)
Layer1 (e.g. WDM)
customer
UNI
access
E-NNI
transport
E-NNI
transport
E-NNI
or other
server

Figure 3.1: Network virtualization architecture showing an example scenario including a Service Provider (SP), a Virtual Network Operator (VNO) network and the physical infrastructure of one or more Physical Infrastructure Providers (PIPs) connected via User Network Interfaces (UNIs) and External Network Network Interfaces (E-NNIs).

### 3.1.3 Service Provider (SP)

The last business role is the Service Provider (SP) who requests a cloud or connectivity service from a VNO. The SP is assumed to have no knowledge and no business interest in operating an own virtual network. It is the interface between the network services realized in the network and the end-users utilizing these services.

There are different amounts of business roles and naming conventions used in the literature. The surveys in [76, 77] define the same business roles as used in this thesis with different names. Some literature [36, 38], additionally defines a fourth business role, namely the Virtual Network Provider (VNP) which acts as a broker between the PIPs and the VNOs. Throughout this thesis we assume the VNP role mainly to be included in the VNO for simplicity purposes, except for Section 7.2, where it is included in the architectural extension discussion to maintain generality.

## 3.2 Resilience Provisioning in Virtual Networks

Resilience is a key property of today's communication infrastructures as it has been discussed in the former chapters. Ever increasing data rates and dependencies of our society and businesses on communication solutions make the high availability of these technologies even more important. Moreover, the communication networks and cloud solutions are also converging [78]. The clouds require high performance network connections to be able to provide satisfactory services to their customers [79, 80]. At the same time, with the new technologies like NFV [47], where the network functions are placed into the cloud, the networks themselves get dependent on the cloud solutions.

In a nutshell, the networks need the cloud to function and the clouds need the network for information exchange and especially to reach the end-customers. Such interdependency requires a conscious coordination between the network and cloud domains. However, currently these domains are often operated by separate entities, making coordinated failure coverage and end-to-end optimization largely impossible. Even only on the network side, end-to-end optimization can be rather difficult since the network operators are not willing to share their topological information. However, to provide a satisfactory performance and reliability to the customers, the services need to be optimized in an end-to-end fashion. For example, reliability plays a crucial role in the decision for adopting cloud services by businesses and is their primary concern according to a survey conducted with over 3700 companies worldwide [3].

Outages do happen: In the past two years, there were many outages, some lasting for hours or days, and they even occurred in the networks and DCs of governments, cities, airline systems, big cloud and network providers, affecting many businesses and millions of users [81]. Besides local causes of outages caused by power outages, fiber cuts, server or router failures, etc., some outages can affect a large area and cause an even larger impact on the businesses and society, for example in case of natural disasters. Communication network and cloud providers need fast and efficient means for recovering from both localized outages and major disasters. Such mechanisms exist today, but when coupled with the problem of separate operation of cloud and network domains, an end-to-end recovery is mostly impossible. This in turn leads to unavoidable outages and/or sub-optimal single domain solutions. One way to overcome this problem is network virtualization that combines the control of network and cloud resources and allows an end-to-end optimization on heterogeneous physical domains.

In this thesis, we propose novel virtual network design models and algorithms, which provide end-to-end resilient solutions. This is achieved via a virtual network design by a VNO, which has an overview of the available resources of the different PIP domains. A VNO is also in the position of operating its virtual network, which is composed of virtual network and IT resources, via a combined controller. For resilience provisioning in such an architecture there are three fundamental alternatives. Resilience can be provided inside the virtual network by considering redundant resources and using re-routing and migration strategies at the design time of the virtual network. The second possibility is delegating the resilience provisioning to the physical layer. In that case, it is sufficient to focus on the performance in the virtual network design and using already resilient virtual resources. These can be for example virtual links, which are mapped on disjoint paths on the physical layer or cloud resources, where the PIP provides a recovery strategy in case of failure, which is ideally transparent to the VNO. As a third option, a combination of these two alternatives can be used, where the cloud resilience is provisioned in the virtual layer, however, network resilience is delegated to the physical layer. This can be especially interesting when a VNO possesses virtual cloud resources or is willing to benefit from the overview it has on the available resources of the different cloud providers but does not have the necessary level of knowledge about network operation or is not willing to do it due to cost or other business related reasons.

All these options bring certain advantages and drawbacks. In the next section, these alternatives are evaluated in an analytical way focusing mainly on the first two to provide a first insight about at which layer it is more beneficial to provision resilience. The quantitative comparisons of different layer resilience solutions can be found in the Chapters 5 - 9 for different requirements and properties. A summary of the quantitative discussion is provided in Chapter 10.

## 3.3 Analytical Analysis of Resilience in Virtual Networks

Resilience in a virtual network environment can be provided either at the VNO or PIP level or at both of them as mentioned in the previous chapter. PIP and VNO experience certain advantages and disadvantages in terms of providing redundant resources and efficient recovery due to their different resource monitoring and controlling capabilities.

Moreover, the optimal recovery strategy depends also on the type of the failure. We differentiate three kinds of failures in a virtual environment, namely software failures, which can cause a VM to either malfunction or completely go down, physical failures (physical node/link failures) and control plane failures. In case of a control plane outage, the data plane will be still functioning and thus a fast recovery is not as essential as in the other cases.

Upon an internal failure of a VM, only the owner and controller of this VM, the VNO, can recognize the failure. Therefore, it is the only one who can react on it. However, a software failure, which causes the whole VM go down or a physical failure will be detected both by the PIP and the VNO, where for VNO physical failures and total VM failures are not distinguishable. In both cases, PIP will be the first one detecting the failure. If the failure is caused by a physical equipment or hypervisor, PIP should react directly by taking the necessary measures and the VM failure should be signaled to the VNO, which should restart its VM. Depending on the contract between the PIP and VNO, it might be the case that VNO also reacts on the failures by rerouting the traffic. In this case, the failure and the recovery action taken by PIP should be signaled to the VNO and the recovery on both layers should be coordinated either by using hold-off timers or failure escalation mechanisms. The only case where a PIP cannot react itself is a catastrophic failure, where the whole PIP goes down. In this case, VNO should either use its already allocated backup resources or request new ones from another PIP for the virtual networks, which were affected.

In the remaining of this section, we will focus on the resilience mechanisms responding to physical failures and complete VM failures, where both VNO and PIP are able to react and we identify their strong and weak points in terms of resource utilization, service level resilience adaptation and complexity.

### 3.3.1 Resource Utilization

In terms of resilience provisioning, the most important advantage of a PIP is that it is the only one having a full knowledge of all its physical and virtual resources. It knows the mapping of the virtual resources to both their physical locations and operating virtual networks. Moreover, it can migrate the virtual resources from one physical location to another, while obeying the requirements of a VNO if there are any, and without affecting the virtual network topologies and disrupting the traffic [82]. All these properties give the ability to the PIP to optimize its network utilization regarding all the virtual networks residing on its network as shown in Figure 3.2(a). It can create back-up resource pools and share them efficiently among the virtual networks by creating special rules depending on the reliability requirements of the virtual networks and the risk groups they share.

The VNOs, on the opposite, have only a limited view on the available virtual resources, i.e. they only have access to the advertised resources of a PIP, and they have no further knowledge about the rest of the network. Therefore, regarding a single PIP domain, a PIP has more knowledge, more freedom and better optimization opportunities at providing redundant resources.

However, even though a VNO has only a restricted view for each of the PIP domains, it has the advantage that it can see the available resources of all of them as shown in Figure 3.2(b). Hence, it can have some opportunities by choosing the backup resources, which are not visible to single PIP entities. It can combine the resources of different PIPs according to its needs and optimize the resilient design of its network in an end-to-end fashion.

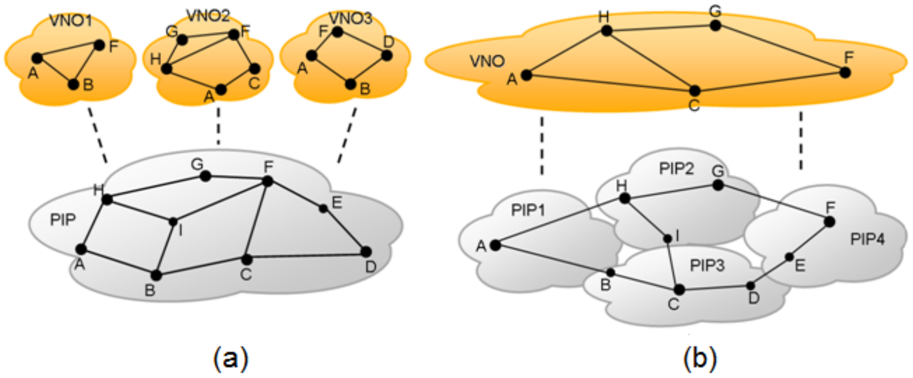


Figure 3.2: Mapping of virtual network(s) on the physical substrate: (a) Various virtual networks mapped on a PIP network, (b) one virtual network expanding over various PIP networks

In the case of resilient cloud resource provisioning, the overview a VNO has on different domains plays even a more important role. A cloud provider would tend to have a geographical distribution of its DCs to both have a protection against geographical failures like disasters as well as to be able to have coverage in a larger area of the network. However, this can lead to the result that the connection to the DR site might not have the desired performance e.g. in terms of latency if this one has a far away location from the service source node at the user side. However, a VNO can select a second DC for protection purposes from any other provider, leading to a better optimization of the end-to-end connection for the cloud services. Moreover, cloud providers having only a single DC are not in the position to offer any resilience solutions in case of complete DC failures.

In both cases, where we have either virtual or physical layer resilience, optimization is done in each layer and domain separately, which leads to suboptimal results for the overall system. This can be overcome only in case that there is only one PIP or one VNO in the virtual network environment or if there is a central unit coordinating the resource allocation for all the PIPs and VNOs.

### 3.3.2 Service Level Resilience Adaptation

Considering service level resilience, the main advantage of a VNO is being the one having knowledge about the actual traffic in the network. Therefore, it can optimize the choice of backup resources and recovery actions on its virtual network accordingly. Moreover, a VNO can adapt the resilience level in its network depending on the needs of the running services, which can be specified by the SP. For some of the services like business-critical services there is a high resilience requirement, whereas for best-effort traffic the resilience mechanisms might not be needed at all.

However, the PIP is rather limited in the sense that it should not have any influence on service handling. Therefore, it cannot optimize its recovery design depending on the traffic flowing in a virtual network or on the quality of service requirements of the individual services. It needs to deal with optimization of resilience at virtual resource granularity instead of at service level granularity. An example to this can be that in case of physical layer resilience, each virtual link on the path of a service, with all the different services running on it, needs to be protected, which can be seen as a link protection. However, in the virtual layer, the services can be protected, if needed, separately via path protection. A more elaborated analysis of this phenomenon in terms of different metrics like virtual network setup cost, network utilization and service latency is provided in Chapter 5.

### 3.3.3 Network Setup and Operation Complexity

As stated before, in this section we mainly focus on the complete VM failures and physical layer failures, where a fast recovery is required. In these cases, the PIP is the one closest to the origin of the failure. This gives to the PIP the advantage of having more knowledge about the failure and being able to react quickly. If the VNOs want to react on the failure as well, a coordination system should be developed, where the failure information is signaled to the affected VNOs.

Moreover, in case a VNO wants to protect its network itself by allocating protection resources and calculating alternative paths and DR sites, it needs to ensure physical disjointness of these resources. Hence, the disjointness information of the virtual resources should be provided to the VNO explicitly at the virtual network setup phase. If resilience is only provided by the PIPs, such information is superfluous and a simple network can be setup.

Finally, one of the key aspects of network virtualization is that several virtual networks can share the same physical substrate, like in the example given in Figure 3.2(a), where all the three virtual networks are sharing the physical nodes A, B and the link between them. Hence, in case of a failure in this shared substrate all of them will be affected. If the VNOs provide resilience for their networks, each VNO has to react separately for the same physical failure. However, if the failure is handled on the PIP layer, the virtual network topology remains unchanged, and therefore, the number of required actions and changes in the system is much lower.

## 3.4 Summary

This chapter aims to form the framework for the remaining chapters of this thesis, where our network virtualization model with all the involved business roles are introduced and the virtual network architecture is described with different possible technological alternatives. Afterwards, we explain why resilience is an important issue in today's networks and cloud infrastructures and how it relates to the usage of network virtualization. We present an analytical analysis by indicating the challenges and opportunities in terms of resource utilization, service level resilience adaptation and complexity that a VNO and a PIP will face when they want to offer resilience for their networks and cloud resources. This chapter provides answers to the following research question:

**Q2.1:** What are the advantages and drawbacks of provisioning resilience in a certain layer in a virtual network architecture? In other words, does network virtualization offer any advantages in terms of resilience compared with traditional resilience provisioning?

This chapter analyzes this question analytically in terms of resource utilization, service-level resilience and network setup and operation complexity. The virtual and physical layer resilience options have both their advantages and disadvantages.

In terms of resource utilization, the physical layer enjoys the complete information about its resources, however, is restricted within its domain in providing resilience. In the virtual layer, resilience design can be performed using an overview of the advertised resources of different physical domains. Therefore, both of these options lack an overall optimization and to find out which one is more efficient under which circumstances further quantitative analysis is necessary, which is presented in the next chapters. In terms of service level resilience, the virtual layer has more benefits since it possesses the knowledge about the services. However, in terms of complexity physical layer resilience is more advantageous due to scalability and signaling issues.

All in all, the decision of resilience layer involves a trade-off. Therefore, the performance of different layer resilience solutions is further analyzed in the next chapters.

## 3.5 Statement on Author's Contributions

This chapter is an extended version of our works in [73] and [72]. The description of the virtualization model and the architectural details presented in this chapter are based on [72], which has been carried out by the author. The work in [73], which has been also carried out by the author, describes the analytical analysis provided in this chapter.

# 4. Virtual Network Simulator Architecture

This thesis is about the resilience aspects of network virtualization, which is seen as a promising concept for the future Internet. As introduced in the former chapter, network virtualization is expected to give rise to new business roles. To be able to develop efficient and realistic virtual network design models, these business roles, their responsibilities and their interactions should be deeply analyzed. This analysis and the understanding about the effects of design decisions and parameter settings in a virtual network ecosystem can be achieved using the means of modeling and simulation. In this chapter, we introduce a new simulation environment, the Java Virtual Network Simulator, which has been developed by the author in the framework of this thesis. In this simulator, a network virtualization ecosystem with all its players and the infrastructures they possess can be modeled, and the performance of new designs for efficient and resilient virtual networks and cloud connections can be assessed.

In the following chapters several virtual network design models for connectivity and cloud services are introduced, which possess different properties. Their performance is then evaluated to answer the research questions, which have been introduced in Chapter 1. All these models are implemented and the simulations are carried out using the simulator, which is presented in this chapter. Thus, this chapter aims to shortly introduce the tools used for the simulations in this thesis to provide a basis for a better understanding of the following chapters.

In this chapter, first the general simulator architecture is introduced. Afterwards, the network modeling and virtual network environment modeling is presented. Finally, an example run of the simulator is described. This chapter is based on our publication [83].

## 4.1 Simulator Description

In this section, the general structure of the simulator is presented. The simulator has six major parts, namely the core with network models, roles in the virtual network environment, topology generators, optimal virtual network design models, routing algorithms and performance comparison methods, which are located in the main package as shown in Figure 4.1. The simulation classes create the virtual network environment, which consists of a given number of different business role types. Each role owns a certain type of infrastructure and can operate those. They can use the topology generators, routing algorithms and virtual network optimization classes to generate and operate their networks.

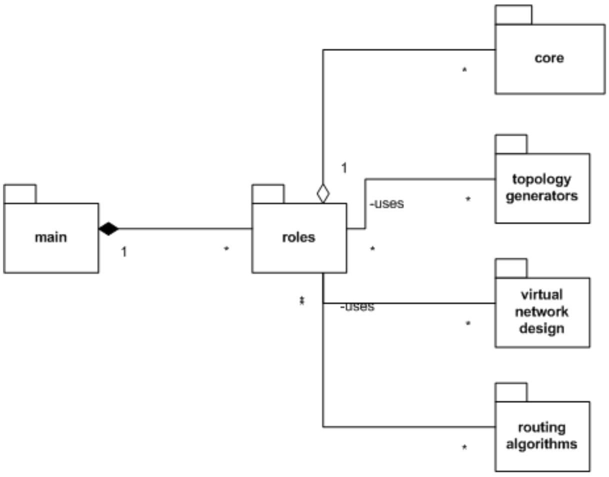


Figure 4.1: General structure of the Java Virtual Network Simulator

Using the simulation methods, propagation delay, cost, resource utilization performance and complexity of different scenarios can be evaluated. In the framework of this thesis, the simulator was used in two different simulation set-ups. In the first one, random virtual networks are generated and they are connected to the cloud providers using different resilience design options, where resilience is provided solely for DC failures. These options are then compared in terms of maximum latency they can guarantee for the services. These models and simulation results are provided in Section 6.2 of this thesis. In the second simulation set-up the virtual networks are designed to provide resilience for a given demand matrix. The modeling is done in the form of linear optimization problems or heuristics. Virtual networks can be designed to serve connectivity or cloud services with end-to-end resilience. Performance of these models is again evaluated using the simulation classes and are presented in the subsequent chapters of this thesis. The simulation classes are designed to provide the results with certain confidence levels and confidence intervals.

Our simulator is developed using the Java programming language. For graph modeling the Java Universal Network/Graph Framework (JUNG) library [84] is utilized. It is a Java library providing some necessary functionalities to model, write/read and analyze graphs and networks. Input files are provided in the Graph Modeling Language (GML) format. For routing the services in the virtual and physical networks, several algorithms are implemented. These algorithms are k-shortest paths[1] and k-shortest disjoint paths. Finally, for modeling the optimization problems, Concert Java library [85] is used, which enables modeling of the optimization problem using the provided Java classes so that the optimization problem can be integrated to any Java program. For the optimization problems, the solver CPLEX version 12.3 [67] is used.

## 4.2 Network Modeling

We model the virtual network environment elements as a multi-layer graph and the relevant interactions between them. Each layer holds information related to itself and about the layers that it is in direct contact with in a distributed manner. Thus, the physical layer has knowledge about the virtual resources residing in its network. A virtual network has the mapping information to the physical layer as well as the routing of the services on itself. In a real world scenario, the level of information sharing would depend on the interface of the different roles. In our model it is assumed that the mapping information

[1] When all simple paths between two nodes are listed in ascending order according to their lengths, k-shortest paths between these two nodes are the first k paths in the list.

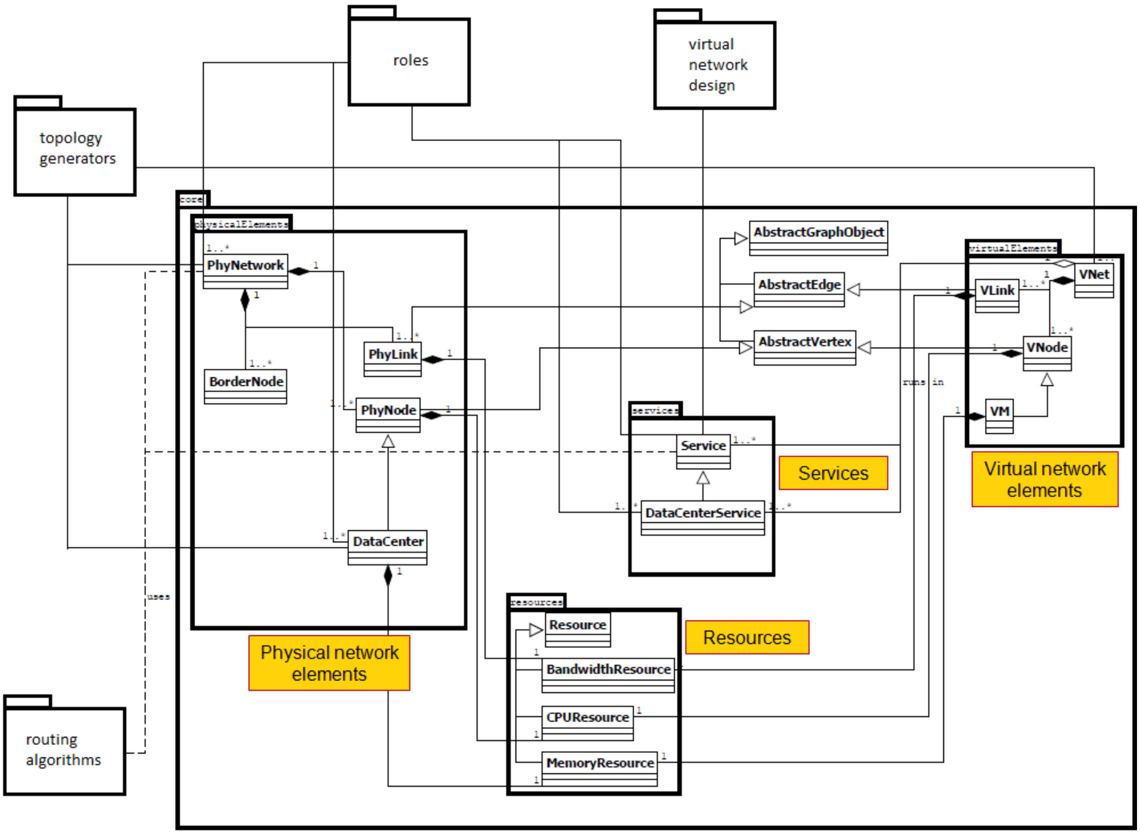


Figure 4.2: The core package: It holds the virtual and physical network classes together with their components and the services running on them.

or at least the necessary characteristics of the virtual resources for optimization purposes are available to the VNO either directly or possibly over certain signaling mechanisms of the PIP for certain purposes like failure recovery. The virtual and physical networks with their components and the services running on them are modeled in the core package as shown in Figure 4.2.

### 4.2.1 Physical Network

A physical network has nodes, links and DCs attached to it, which are modeled using the *PhyNode*, *PhyLink* and *DataCenter* classes, respectively. In our model, the network and service information is saved in a distributed way. All physical resource objects hold information related to their properties and also about the virtual layer resources mapped on them. This enables an easy control of interactions of the different layers. A PIP can reserve upon the request of a VNO, certain virtual resources on a physical network resource using the related data saved on this resource object. All resources are defined by the parent class *Resource*, where a PhyNode has a certain amount of *CPUResource* and a PhyLink has a certain amount of *BandwidthResource*. A DataCenter, which is a subclass of PhyNode, possesses additionally *MemoryResource*s.

A physical network is modeled with the class PhyNetwork<N,E> and it extends the *SparseMultiGraph*<N,E> class from the JUNG library. It contains physical nodes and links. The network resource management is done using the inherited methods. The additional information it holds is about its owner PIP and the list of the border nodes with other networks, which are defined as the closest nodes of the two networks.

The physical network also has the functionality to provide the k-shortest paths and disjoint path pair information between two given physical nodes. These methods are used by the

PIP in the case if the information is not already stored in the nodes. The distributed data saving in the beginning of the program for all the network resources enables faster continuous simulation runs. Finally, the physical network can also provide its minimum, maximum and average node degrees using the corresponding methods.

### 4.2.2 Virtual Network

A virtual network is modeled similarly to a physical network. It has virtual nodes and links and also virtual machines, which are modeled as a subclass of a virtual node. The virtual nodes and links hold the information regarding their properties, mappings and usage by services. A virtual link can request some BandwidthResources, a virtual node some CPUResources and a virtual machine additionally some MemoryResources. Moreover, a virtual link also holds the information regarding the observed propagation delay on itself due to its physical mapping.

A virtual network is modeled with the class VNet<V,E> and it extends the SparseMultiGraph <V,E> class. It contains virtual nodes, *VNodes*, and links, *VLinks*. The network resource management is done using the inherited methods. The additional information it holds is about its owner VNO, the physical network it is mapped on, its cloud connections, and the list of the unicast and cloud services running on it. If requested, the cloud connection information is saved in the *DataCenterConnection* object, which contains information about the primary and backup DC sites and their connection paths to the virtual network.

The services are modeled with the *Service* class. A service has a target and a source node, which are both VNodes. Moreover, it holds the information related to its requested link and node resources. Finally, it also has the fields for its routing both in the virtual network and the physical network for both primary and working paths and the corresponding end-to-end delay. A *DataCenterService* has target nodes for primary and DR site DC fields with corresponding routing information to its source node. A virtual network can also provide the information about the routing of the services running on itself, its topological connectivity and the mapping status of all its nodes and links. Moreover, it provides physical disjointness information about its resources, too.

## 4.3 Virtual Network Environment Modeling

A virtual network environment consists of different roles, namely of the VNOs and the PIPs and of the functionalities to manage these roles. Note that multiple VNOs and PIPs can co-exist in a virtual network ecosystem and interact with each other. The interactions can be both vertical and horizontal, where the former one is the expected business relationship for having and operating virtual networks and the latter is for peering purposes. The roles package containing the business role classes is depicted in Figure 4.3 and in the following subsections the models of these business roles and the general functionalities are introduced.

### 4.3.1 VNO

The VNO class models the VNO business role, which can operate several virtual networks, and hence holds a list of the virtual networks belonging to and operated by this VNO. Moreover, it also has a field to keep the information about the advertised resources of all the available PIPs. The resources of the NetworkPIPs are kept as a merged single physical network and the available resources in the DCs are kept as a list of DataCenter objects. A VNO can manage its virtual networks by loading new ones from a given file, deleting them, writing them to a file and generating random or optimized virtual networks using the virtual network generators or the optimization models. To reserve resources for a virtual network, the VNO triggers the corresponding PIP(s).

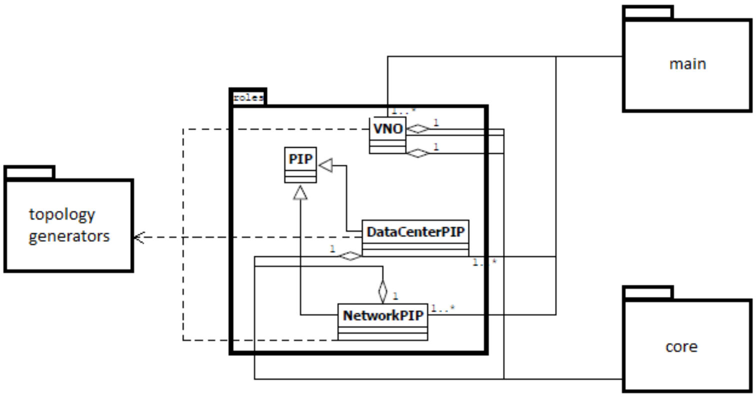


Figure 4.3: The roles package holding the classes modeling the business roles in a virtual network environment

During the virtual network design process, the available network and IT resources from several NetworkPIPs and DataCenterPIPs are advertised to the VNO in the form of a merged physical graph topology as well as a list of DataCenter objects with their available resources. The routes of the services are computed for both the normal operation and failure cases if resilience is provided in the virtual layer. Otherwise, a simple virtual network is designed using resilient virtual resources, where resilience is provided by the PIPs. According to the requirements of the service requests, the corresponding optimization models or heuristics, which are part of the virtual network design package as introduced in Section 4.5, are used to determine the routing of the services and the necessary virtual resources to be rented from the PIPs.

### 4.3.2 NetworkPIP

The NetworkPIP class extends the PIP class. It owns a physical network and hence has a PhyNetwork with the corresponding management methods. Those methods are used to load the network from a file or write to a file, to set or reset it and to generate a new network randomly using specific algorithms, which are described in detail in Section 4.4. In a real world scenario, a nPIP is expected to have a central database to keep the up-to-date information about its physical network. For the sake of simplicity we model the information database as a distributed system, where the PIP has direct access to all the information held in its network resources using the implemented management methods. A NetworkPIP can reserve/release virtual resources on its physical resources upon request of a VNO. The resources are checked for availability and if enough resources are available they are reserved on the requested physical resource and the information is returned to the VNO by changing the mapping setting of the virtual resources.

### 4.3.3 DataCenterPIP

The DataCenterPIP class is also a subclass of the PIP class. This one is assumed to have only DCs and therefore has a list of the DataCenters it owns. It can generate a random DC list using the DCGenerator, which is introduced in Section 4.4, load DCs from an input file or write the existing DC list to an output file using the general file reading/writing classes.

If DC resilience is to be provided in the physical layer, the primary DC for a virtual network or service is selected by the VNO operating this virtual network. This information is then given to the DataCenterPIP owning the corresponding DC, and this one can choose the DR site DC(s) from its domain according to different internal strategies. As possible internal strategies of a DataCenterPIP, the following are modeled and implemented in the tool:

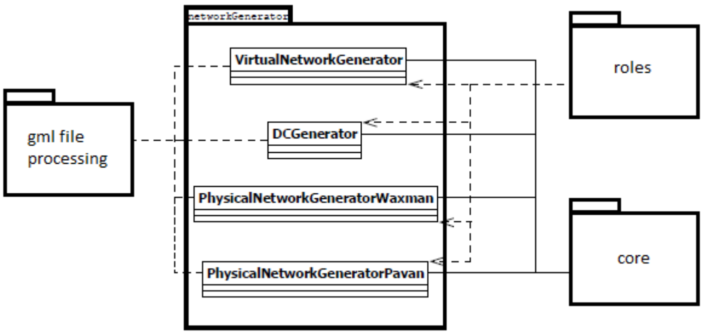


Figure 4.4: The network generator package holding the algorithms for random physical and virtual network generation

- **Load Balancing:** The DataCenterPIP chooses the DC(s) from its own domain as the DR site DC(s), which have currently the minimum load.
- **Shortest Distance:** The DataCenterPIP chooses the DC(s) from its own domain as the DR site DC(s), which provide the shortest connection path(s) to the primary DC.

Two different resilience levels are considered in our implementation, namely protection only against DC failures or also against network failures. In both cases, the DataCenterPIP is responsible for reserving the DR site DC(s) and the inter-connection path(s) of the primary and DR sites, which is used to redirect the traffic to the DR site(s) in the physical domain in case of failure. If network resilience is also to be considered, two or more disjoint physical paths are reserved to inter-connect the primary and DR sites. Note that these paths can be reserved directly by the DataCenterPIP assuming a peering with the corresponding NetworkPIP(s) owning the necessary physical network.

### 4.3.4 VirtualNetworkEnvironment

This class is designed to manage the roles in the virtual network environment. It holds the lists of VNOs, NetworkPIPs and DataCenterPIPs and allows to add and delete the roles. Moreover, one can load the roles from files using the appropriate methods, where the networks are specified in the input files. Additionally, this class provides the functionality of determining the border nodes of two physical networks and generating an overall graph by merging the networks and DCs of all the available PIPs. Finally, it also offers a method to calculate the k-shortest paths and disjoint path pairs between all the nodes and populate the nodes with this information.

## 4.4 Random Topology Generators

In this section the topology generators for physical networks, virtual networks and DC sets are introduced. These generators can be called from the corresponding roles to add networks and DCs to their infrastructure as shown in Figure 4.4.

### 4.4.1 Physical Network Generator

There are two classes for generating physical network topologies. They are called by the NetworkPIP when a new random physical network is required. The first one is based on the Waxman probability [86] and the second one is based on the algorithm given by Pavan et al. [87], which uses [86] as a basis and develops a more realistic model. These both classes have only static methods, which allow the user to create new physical nodes, physical links and finally random physical networks. For both cases the topological input parameters like the area, number of nodes, node and link resources and the connectivity of the network should be specified. In case Waxman probability is used, the alpha and beta variables, which are part of the algorithm described in [86], should be also specified. If the class based on the algorithm in [87] is used, some additional information like the number of the regions the total area should be divided in, minimum distance between nodes, minimum node degree and maximum node degree should be also provided.

### 4.4.2 Virtual Network Generator

This class has also only static methods, which allow the VNO to create new virtual nodes, virtual links and random virtual networks. The optimal resilient virtual networks are created via another package. The random virtual networks can be generated to be already mapped on a given physical network or without any mapping specified. For virtual network generation the input parameters for the number of nodes, mapping choice on the chosen physical network, the requested resources and connectivity of the network should be defined. The generated virtual networks can be chosen to be just connected, bi-connected or complete graphs.

### 4.4.3 DataCenter Generator

The DCGenerator class has only static methods, which can be used by a DataCenterPIP under different circumstances. For generating random DCs information about their network connection nodes, capacities and how many DCs are going to be generated should be provided. There are three different options for DC generation, namely the DCs can be located in a physical network randomly, according to a specified inter-DC distance or such that each DC is located in a different availability region for disaster recovery.

## 4.5 Virtual Network Design Models

This is the package, which is the heart of the whole simulator tool. There are different classes for virtual network design models with various properties. The virtual network design models are grouped into subpackages according to their properties. Both the unicast and anycast packages hold the classes to perform virtual network design without resilience as well as with resilience either in the physical or in the virtual layer. Moreover, the models from prior art, which possess certain limitations as described in Chapters 5 and 6, are also implemented in the corresponding packages. There are also models with additional properties like offering shared protection, QoS guarantees and enhanced resilience, which are implemented in their corresponding packages and described in detail in the following chapters of this thesis.

In all these models, the methodology described in Figure 4.5 is followed. The abstract problem of virtual network design is first modeled mathematically. These models are presented in detail in the following chapters. The mathematical models, the MILPs, are then implemented using the IBM Concert Java library as mentioned in Section 4.1. This models are solved using the CPLEX solver as part of the main simulation as described in the next section. Solving the models yields the value of the objective function, which can be e.g. minimizing the virtual network setup cost or minimizing the service latency, as well as the values of the variables, which result in the optimal value of the objective function. The variables are used to describe the virtual network topology, e.g. if a virtual link/node is part of the resulting virtual network or not, and the service routing within the virtual network, e.g. if a service is using a certain link/node in its route or not. Therefore, this information can be used to generate a virtual network graph together with the services running on it accordingly. Afterwards, certain measurements and tests are performed on

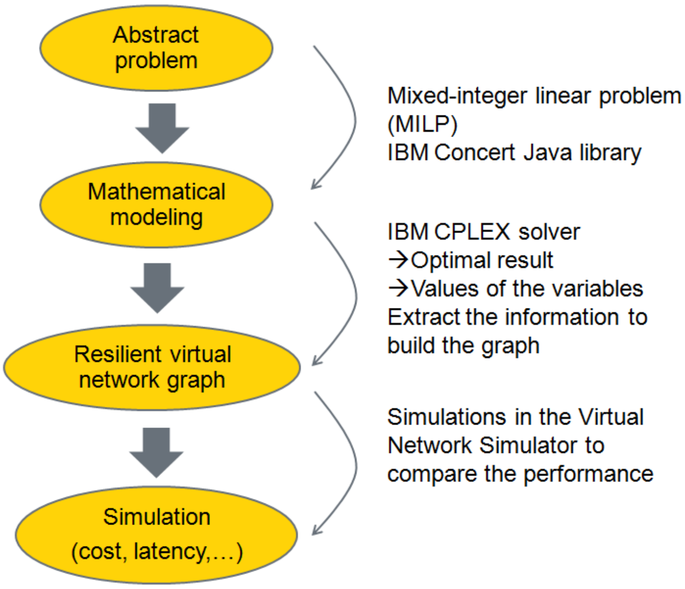

Abstract problem
Mathematical modeling
Resilient virtual network graph
Simulation (cost, latency,…)
Mixed-integer linear problem (MILP)
IBM Concert Java library
IBM CPLEX solver
→Optimal result
→Values of the variables
Extract the information to build the graph
Simulations in the Virtual Network Simulator to compare the performance

Figure 4.5: The methodology used in this thesis to reach from the abstract problem of how to design resilient virtual networks to model, implement and evaluate the proposed algorithms

this topology in the framework of the main simulator, as described in the following section, to evaluate the performance of the proposed models in terms of virtual network setup cost, service latency, resource utilization and complexity.

## 4.6 Virtual Network Design Performance Simulation

In this section the main simulator classes are introduced. Note that using the simulator classes of the tool many different simulations can be realized. We have two type of simulator classes implemented, one focusing on the connectivity services and one on cloud services. In the following the details of the latter is described since it is more comprehensive. The simulation aim can be set as the virtual network setup cost, maximum or average service latency, network and/or IT resource utilization and number of virtual links/nodes used in the generated virtual network as a result of the optimization problem.

### 4.6.1 Example: Latency Performance Simulation for Cloud Services

All the simulation classes are designed to provide results with certain confidence levels. In general 95% confidence level with ±5% or ±1% confidence interval is used. However, the confidence level is parametrized and can be adjusted according to the current needs.

In the simulations, the virtual network environment is modeled to have several NetworkPIPs and DataCenterPIPs and a single VNO requesting and designing resilient optimal virtual networks. For each virtual network a uniform demand matrix is used, where there is an anycast demand originating from each of the specified source nodes. The aim of the simulation is designing resilient virtual network topologies for cloud services having either physical or virtual layer resilience and comparing them in terms of maximum propagation delay of the cloud services in the virtual network in each design option.

There are three levels of simulation as presented in Figure 4.6. As mentioned above in the simulation environment there are several NetworkPIPs owning the physical networks. These networks can be either provided by the user or can be generated randomly using the topology generator. In the second case, for each generated physical network, the simulation

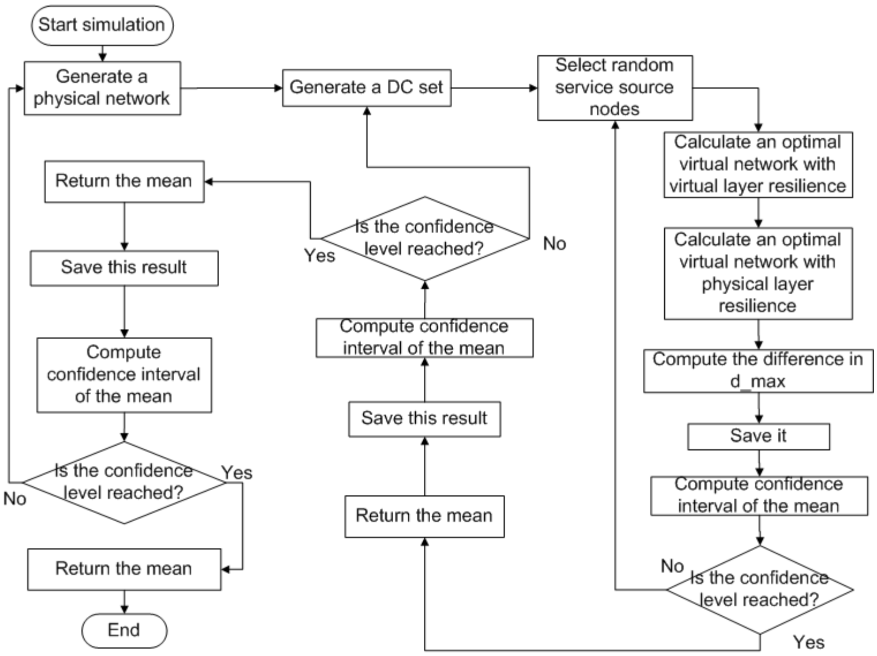

Start simulation
Generate a physical network
Generate a DC set
Select random service source nodes
Calculate an optimal virtual network with virtual layer resilience
Calculate an optimal virtual network with physical layer resilience
Compute the difference in d_max
Save it
Compute confidence interval of the mean
Is the confidence level reached?
No
Yes
Return the mean
Save this result
Compute confidence interval of the mean
Is the confidence level reached?
Yes
No
Return the mean
Save this result
Compute confidence interval of the mean
Is the confidence level reached?
No
Yes
Return the mean
End

Figure 4.6: Description of the simulator class for cloud services with simulation aim of maximum end-to-end service latency

is run once and the results are calculated. This procedure is repeated until the results are in a certain confidence interval with a given confidence level. After the generation of the physical network, the methods realizing the inner simulation loops are called.

The second level of the simulation is for the DC sets. The number of the DataCenterPIPs and the number of DCs per DataCenterPIP, the placement strategy of the DCs, the number of availability regions, in which the overall network should be divided, and the minimum distance between the DCs should be specified as input parameters. The second level simulator is called by the upper level for a given or random physical network. For the this physical network, random DC sets are generated using the methods described in Section 4.4. For the physical network and the generated DC set, the third level simulation is called, which is described below. The process of generating a new DC set and calling the third simulation loop continues until for the results of the third level simulation for different DC sets a certain confidence interval with a given confidence level is reached. Note that if simulations are done solely for connectivity services, this second simulation loop with the DC sets is not used.

The third and last simulation level calculates the resilient virtual networks and performs certain tests on them. The physical network and the DC set are provided from the higher simulation levels as inputs to this simulation level. In this simulation loop, resilient virtual networks are generated, which offer end-to-end resilience for cloud services either at the physical or at the virtual layer. Then, for both of the resulting virtual network topologies the maximum propagation delay occurring in these end-to-end systems is calculated and compared. The process of generating new virtual networks and performing the comparison is continued until the required confidence level is reached for the mean of the latency result. As stated before, this result is returned to the second level simulator, which continues to run until its confidence level for different DC sets is reached. This one returns the result then to the last level, which calls the second level and implicitly the third level simulation for different physical networks until the overall results are in a certain confidence interval with the required confidence level.

## 4.7 Summary

This chapter introduces the Virtual Network Simulator architecture. Such a simulation model is very important to assess the effects of different parameters in a virtual network ecosystem and to be able to try new methods for virtual network design and routing. Our tool focuses on resilience design in this ecosystem, since it is one of the most critical design aims in today's and future networks and offers powerful simulation classes for the performance evaluation of the different resilience models including the ones for cloud connections. The tool builds a basis for analysis in a virtual network ecosystem with various roles and their interactions and it can be easily further extended for the analysis of different performance metrics.

The aim of this chapter is forming a basic understanding of the simulation setup, which is used in the following chapters for the implementation and evaluation of the presented optimization models and heuristic algorithms. We present here an overall and concise description of the simulator tool, and further details of implementation are explained in the corresponding chapters if necessary.

## 4.8 Statement on Author's Contributions

This chapter is an extended version of our work in [83]. The simulator described in [83] has been developed and implemented by the author. Since the paper describes an earlier version of the simulator, the Sections 4.5 and 4.6 have been updated to match the current version of the simulator tool. Moreover, the figures throughout this chapter are added, which have not been used in the paper due to space limitations.

# Part III

# Resilient Virtual Network Design Models

# 5. Optimization Models for Resilient Virtual Network Design

In the former chapters the details of the virtual network environment have been introduced. There are different business roles responsible for various tasks. The PIP is the owner of the physical substrate and is advertising its available virtualized physical resources to the VNOs, which are then in the position to select the resources they want to use and order the setup of a virtual network by the PIP(s) using these selected resources. Their resource selection depends on various parameters and constraints, where the main aim is satisfying the connectivity service requests of the SPs in a cost-efficient manner.

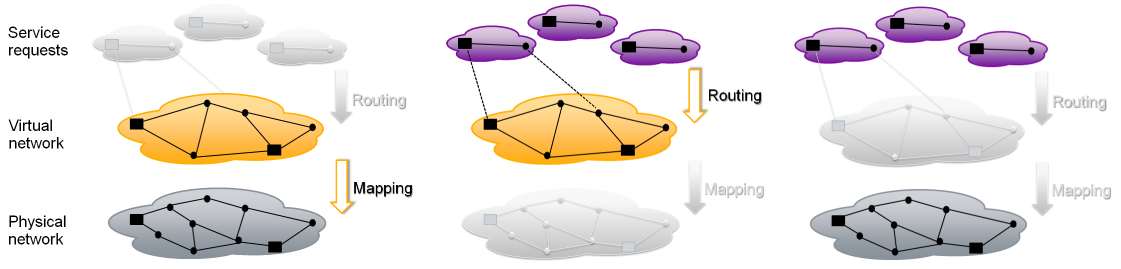
Resilience provisioning, which is a key issue for today's and future networks, adds another complexity to the virtual network design due to the separation of the physical and virtual layers, and hence, limitation of the available information. At this point, the second important question raises, namely at which layer to provide resilience.

In this scenario, together with the resilience considerations, the main open question is how a VNO can perform the virtual network design using the inputs and constraints coming from the physical and service layers. To obtain optimal solutions, the service routing and virtual resource selection or virtual network mapping have to be performed simultaneously. Moreover, there are two fundamental alternatives for provisioning resilience, namely in the virtual or in the physical layers, and both of these options need to be elaborated.

In this chapter, we will first explain why resilient virtual network design is an open question and what kind of partial or similar answers are available in the literature in Section 5.1. Afterwards, we will describe the proposed methods in detail, and provide a performance evaluation both comparing them with the prior approaches and answering the question of where to provision resilience in Sections 5.2 and 5.4, respectively. These sections are an extended version of [88]. The differences of these sections with [88] are listed in Section 5.8. In Section 5.5, a cost and resilience premium analysis is presented for the proposed models. Section 5.6 introduces a general heuristic framework for resilient virtual network design. Finally, Section 5.7 concludes this chapter summarizing the main outcomes.

## 5.1 Related Work and Our Contributions

In this section, the existing literature in the area of virtual network design is discussed and the differences with the requirements of the open problem are elaborated. The related



(a) Overlay networks (b) Resilient VPNs and VN embedding (c) Resilient virtual networks

Figure 5.1: Virtual network design problem and different approaches from the literature, (a) In case of overlay networks, the virtual network topology is already given and mapped onto the physical network and the aim to reliably route the services within this network. (b) In the area of VPNs and virtual network embedding, the virtual network topology is again provided and the question they answer is how to map it reliably onto the physical network. (c) However, in case of a virtual network architecture, the virtual network topology is not given a-priori and has to be designed by the VNO using the available resources of the physical layer and the service requests as its input.

work about the analysis of resilience cost and the heuristics for virtual network design is presented in the corresponding sections.

In a virtual network environment, there is a three layered architecture composed of the physical networks of the PIP(s), virtual networks of the VNO(s) and the service requests coming from the SPs(s). The open problem is designing an efficient resilient virtual network, which should be mapped onto the physical substrate and where the services are routed within the virtual network as shown in Figure 5.1. Regarding the virtual network design there are mainly two types of works in the literature as described in the following.

The first group of literature is on routing the services in a virtual network according to given criteria like QoS or availability requirements [89, 90]. It is assumed that the virtual network topology is already existing and mapped onto the physical substrate as shown in Figure 5.1a. In [89], the virtual network is an overlay topology on top of the physical substrate and the aim is measuring the quality of the virtual links to sense failures and route/re-route the traffic accordingly. This method can bring benefits in terms of reliability compared with traditional routing protocols like Border Gateway Protocol (BGP) and Open Shortest Path First (OSPF), however, it does not provide an answer to the question of how to design a virtual network. It solely uses a given virtual topology to improve the quality of routing. In our problem space, we need to have a method to design the virtual network topology, which can offer the required QoS and reliability, where for the setup of each virtual network resource a VNO needs to pay a certain fee to the PIP(s). In [91], the authors deal with the virtual network design problem for the case of the overlay networks by minimizing the cost of the overlay network. However, they do not consider resilience and like many other literature they use direct shortest path mappings. In [92], resilient VPN designs are realized but again assuming direct mapping of the virtual links on shortest physical paths. We extend this approach by allowing several mapping choices for a virtual link and show that our approach outperforms the shortest path mapping model in terms of feasibility and delay performance.

The second type of work in the literature offers solutions for the mapping of a virtual network onto the physical substrate [93, 94] including the mapping of survivable virtual networks [95, 96]. However, all of these works assume that the virtual topology is already given - possibly together with the service routing, and the problem they address is the sub-problem of how to embed this virtual topology onto the physical network as shown in Figure 5.1b. In case of survivable mapping, the virtual network has to be even bi-connected so that the mapping can be realized at all. Our work does not have such a limitation and designs a virtual network with the given requirements. Thus, there is extensive work available for mapping a given virtual network on the physical substrate and routing a set of services in the virtual layer, which is already mapped onto the physical substrate. This is however a sub-optimal solution due to the limitation of the service routing and does not provide any answer to the virtual network design problem since the topology is assumed to be known a-priori, which is not possible when a VNO designs a new virtual network. To the best of the author's knowledge, this thesis, based on [88], is the first work, which considers both the virtual network embedding and service routing simultaneously to realize cost-efficient and latency-optimized resilient virtual network designs as shown in Figure 5.1c.

Finally, the literature in the area of multi-layer resilience [97, 98, 99] can be seen as relevant considering the different layers of a physical network as the physical and virtual topologies. Multi-layer resilience mainly deals with the interaction between these layers and cross-optimization of these layers using their interdependencies. The main difference of this type of work compared with virtual network design is that it completely ignores the vertical division of the ISPs into different business roles. This division creates the above-mentioned three layered structure, where the design of the virtual network does not only aim to have a bandwidth efficient routing as in a multi-layer network case but has to consider the business relationship of the VNOs and the PIPs. Note that the physical network of a PIP can be itself a multi-layer network, and hence, the methods from the multi-layer resilience literature can be directly applied there.

In conclusion, the related work from the areas of overlay networks, VPNs and multi-layer network resilience falls short in answering the open question of how to design a resilient virtual network in a VNO-PIP business relationship scenario. The main contribution of this chapter is providing an answer to this question in the form of an optimization problem. The proposed model covers resilience provisioning options in both layers. The performance and different properties of these models are in depth evaluated in the following sections. A general framework for virtual network design heuristics is also provided.

## 5.2 Virtual Network Design Model

In this section, virtual network design models are introduced, which allow simultaneous optimization of virtual network embedding and service routing. As explained in the previous section, this property is crucial for the virtual network design by a VNO. The input a VNO receives from the SPs and PIPs is translated into a set of requested connection services and the physical network topology graph. The virtual network design is modeled as a MILPs and the optimization is performed for minimizing the maximum latency observed in the virtual network or the cost of the virtual network. This section provides the details on the main model without any resilience considerations. In the following section resilience designs for the virtual network layer (VNO-Resilience) and for the physical network layer (PIP-Resilience) are introduced, which are defined as additional constraints and/or special inputs to the main model.

### 5.2.1 Main Model without Resilience

In the *Main Model* the virtual links are mapped onto single paths in the physical network and the services are routed in the virtual network on $i \in \{1, .., k\}$ routes. The *service nodes*, i.e. the end-nodes of the given services, are directly used as the virtual nodes of the resulting

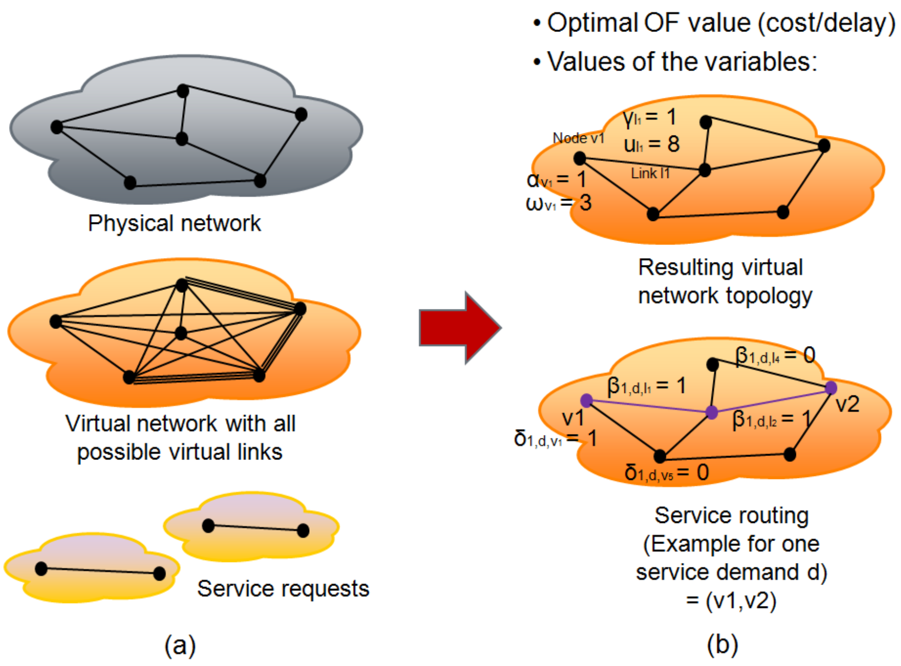


Figure 5.2: Inputs and outputs of the optimization problem, (a) The physical network topology, virtual link and node candidates and the connection service requests are provided as inputs. (b) The outputs of the MILP are the optimal value of the objective function, virtual network topology and the service routings within the virtual network topology.

virtual network. The virtual links have k-shortest paths[1] mapping possibilities. However, to maintain linearity instead of using one virtual link with several possible mappings, we generate a new virtual link for each mapping and add it to the list of all possible virtual links. The result of the optimization problem is a virtual network topology, which consists of only the links and nodes that are used to route any of the given services.

In conclusion, the MILP takes as input all virtual link and node candidates, with their mapping, properties (e.g. end-to-end latency, cost etc.) and limitations (e.g. capacity limitations), and the set of connection service requests as shown in Figure 5.2. For each virtual link its cost, mapping and end-to-end latency is provided. Similarly, for each virtual node its cost and its mapping are given. For the connection service requests, the end-nodes and the required bandwidth and node resources are specified. The MILP returns a optimal value for the objective function as well as the values of the optimization model parameters, from which the information about the virtual network topology and service routing can be extracted as given in Figure 5.2.

#### 5.2.1.1 List of Sets, Parameters and Variables

In the following, the sets, parameters and variables used in the main model as well as in the resilience design models are briefly introduced.

- *Sets:*
  - $V$: Set of the all virtual node candidates
  - $L$: Set of the all virtual link candidates
  - $D$: Set of the requested connectivity services
  - $E_l$: Set of the endpoints of link $l \in L$
  - $Z$: Set of virtual links $(j, k) \in L^2$, which share at least one physical edge

[1] When all simple paths between two nodes are listed in ascending order according to their lengths, k-shortest paths between these two nodes are the first $k$ paths in the list.

- *Parameters:*
  - $b_d$: Requested bandwidth for the service $d \in D$
  - $n_d$: Requested node resources for the service $d \in D$
  - $t_l$: Physical length of link $l \in L$
  - $\lambda_l$: Fixed setup cost for having a new link $l \in L$
  - $\theta_l$: Setup cost per unit capacity for link $l \in L$
  - $\mu_v$: Fixed setup cost for having a new node $v \in V$
  - $\eta_v$: Setup cost per unit capacity for node $v \in V$
  - $r_{\text{PIP}}$: Resilience premium for providing resilience for a certain virtual network resource in the physical layer
- *Variables*
  - $\beta_{i,d,l}$: Binary variable taking the value of 1 if the link $l \in L$ is used for the $i^{th}$ route of the demand $d \in D$, 0 otherwise
  - $\delta_{i,d,v}$: Binary variable taking the value of 1 if the node $v \in V$ is used for the $i^{th}$ route of the demand $d \in D$, 0 otherwise
  - $\gamma_l$: Binary variable taking the value of 1 if the link $l \in L$ is in the resulting virtual network, 0 otherwise
  - $\alpha_v$: Binary variable taking the value of 1 if the node $v \in V$ is in the resulting virtual network, 0 otherwise
  - $u_l \in [0, \infty]$: Used capacity on link $l \in L$
  - $\omega_v \in [0, \infty]$: Used capacity on node $v \in V$

#### 5.2.1.2 Objective Function

There are two objective functions defined for different optimization objectives, namely virtual network cost minimization and delay minimization. Virtual network setup cost is the price a VNO needs to pay to a PIP for the setup of a virtual network with a set of selected virtual network resources. We have selected the virtual network setup cost as one of the optimization objectives since the cost is mainly the most important driver for the businesses. Besides the cost, a certain level of performance a VNO can guarantee to its customers is also of high importance to ensure customer satisfaction. Service performance optimization is realized with the use of the delay minimization objective function.

The cost of the virtual network constitutes of the virtual link cost and the virtual node cost, where each of them has again two parts, namely the fixed setup cost of having a new link or node in the virtual network and the capacity dependent cost (per unit capacity) depending on the requested capacity on that link or node. To achieve simplicity in the PIP-VNO business relationships, a linear cost model is assumed. This cost model is shown in Figure 5.3. A deeper discussion on this cost model is provided in Section 5.5. In cost minimization, the total cost of the virtual network is minimized as given in (5.1).

$$\min \quad \Big( \sum_{l \in L} (\lambda_l \gamma_l + \theta_l u_l) + \sum_{v \in V} (\mu_v \alpha_v + \eta_v \omega_v) \Big) \tag{5.1}$$

In propagation delay minimization the total length of the routes for each service is minimized. In this basic model, we only consider the propagation delay in the physical path as the latency metric for a service since the network is assumed to be designed for non-full load conditions. Thus, the queuing delay is sufficiently low and the main latency is caused by the propagation of the signal over physical distances. Expression (5.2) shows the objective function for delay minimization of the services.

$$\min \quad \sum_{d \in D} \sum_{i \in \{1,..,k\}} \sum_{l \in L} \beta_{i,d,l} \, t_l \tag{5.2}$$

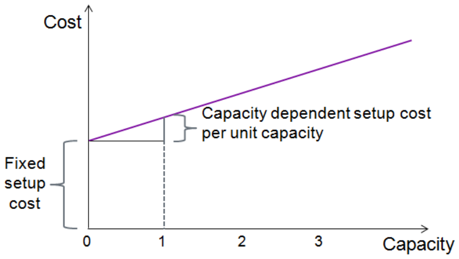


Figure 5.3: The cost model of all virtual network resources has the same structure. It has a fixed cost component for the setup of this new virtual resource and the unit capacity cost component depending on the requested capacity on this resource.

#### 5.2.1.3 Constraints

The constraints for the main model are given in the following. Equation (5.3) is the non-splittable flow conservation constraint. Equation (5.4) makes sure that a node is flagged as "used" for a service if it is the source or the target of that service. Equations (5.5) and (5.6) state that a virtual link or node is part of the resulting virtual network if it carries the traffic of any service, respectively. Finally, (5.7) and (5.8) are the constraints for link and node capacity, respectively.

$$\sum_{l:v\in E_l} \beta_{i,d,l} = \begin{cases} 1 & \text{if } v = s \text{ or } v = t \\ 2\delta_{i,d,v} & \text{otherwise} \end{cases} \quad \forall d=(s,t)\in D,\ v\in V,\ i\in\{1,..,k\} \tag{5.3}$$

$$\delta_{i,d,v} = 1 \quad \forall d=(s,t)\in D,\ \forall v\in(s,t),\ i\in\{1,..,k\} \tag{5.4}$$

$$\gamma_l \geq \beta_{i,d,l} \quad \forall l\in L,\ \forall d\in D,\ \forall i\in\{1,..,k\} \tag{5.5}$$

$$\alpha_v \geq \delta_{i,d,v} \quad \forall v\in V,\ \forall d\in D,\ \forall i\in\{1,..,k\} \tag{5.6}$$

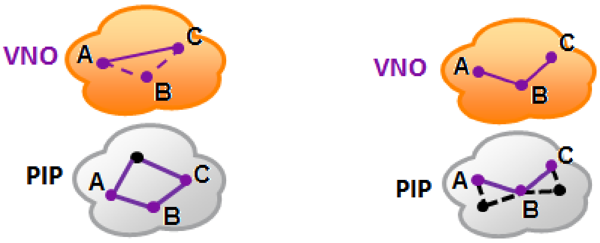
$$u_l \geq \sum_{i\in\{1,..,k\}} \sum_{d\in D} \beta_{i,d,l}\, b_d \quad \forall l\in L \tag{5.7}$$

$$\omega_v \geq \sum_{i\in\{1,..,k\}} \sum_{d\in D} \delta_{i,d,v}\, n_d \quad \forall v\in V \tag{5.8}$$

In case the delay minimization objective function is applied, the virtual link and node usage indication variables $\gamma_l$ and $\alpha_v$ are not minimized, and hence, they do not possess any upper bound unlike for the cost minimization case. Therefore, it is necessary to introduce their bounds as given in (5.9) and (5.10), which ensure that a link or node, which is not used by any service, is not included to the resulting virtual network topology.

$$\gamma_l \leq \sum_{d\in D} \sum_{i\in\{1,..,k\}} \beta_{i,d,l} \quad \forall l\in L \tag{5.9}$$

$$\alpha_v \leq \sum_{d\in D} \sum_{i\in\{1,..,k\}} \delta_{i,d,v} \quad \forall v\in V \tag{5.10}$$



(a) VNO-Resilience (b) PIP-Resilience

Figure 5.4: Simplified example of the resilience design models for a single service and $k$=2. The solid lines show the working paths and the dashed lines show the protection paths. The used resources of the physical layer are presented in violet. (a) VNO-Resilience: Each virtual link is mapped on a simple physical path. Resilience is provided by the VNO by routing each service on a working and (at least one) protection path, which are physically disjoint. (b) PIP-Resilience: Resilience is provided by the PIP and therefore each virtual link is mapped on working and (at least one) protection paths in the physical layer. It is sufficient to have a simple path routing within the virtual network.

## 5.3 Resilience Models

Resilience is a key feature for today's networks and will be of even higher importance in the future networks due to increasing dependency of businesses and private applications on communication services and due to ever increasing data rates. In a virtual network environment, there are two fundamental alternatives of providing resilience for connection services. Resilience can be provisioned either in the virtual network layer (VNO-Resilience) or in the physical network layer (PIP-Resilience) by the corresponding business roles. This section introduces the necessary changes compared with the main model in order to provide resilience in each corresponding layer.

### 5.3.1 VNO-Resilience

For VNO-Resilience, $k$-1:1 protection routing is used in the virtual layer, where the working and protection paths of a service have to be physically disjoint. To provide resilience additional diversity constraints are introduced to the model. The constraint given in (5.11) ensures that the virtual working and protection paths of a service do not contain any two virtual links, which share common edges in the physical layer. Equation (5.12) provides node-diversity, where the working and protection paths are not allowed to share any nodes other than the end-nodes. In case of physical link or node failures, the affected services are re-routed by the VNO on their pre-calculated protection paths. A simplified example of VNO-Resilience is shown in Figure 5.4a for a single service routing and $k = 2$.

$$\beta_{i,d,l} + \beta_{j,d,m} \leq 1 \quad \forall d \in D,\ (l,m) \in Z\ (i,j) \in \{1,..,k\}^2 \tag{5.11}$$

$$\delta_{i,d,v} + \delta_{j,d,y} \leq 1 \quad \forall d = (s,t) \in D,\ (v,y) \in V \setminus \{s,t\}\ (i,j) \in \{1,..,k\}^2 \tag{5.12}$$

### 5.3.2 PIP-Resilience

In case of PIP-Resilience, providing resilience is the responsibility of the PIP(s). The services are routed on single paths in the virtual network layer, where each virtual link is mapped on $k$ disjoint physical paths in the physical layer. The disjointness criteria can be defined as link-disjoint or node-disjoint. For PIP-Resilience, the main model is directly applied where the number of virtual routes is set to 1. However, instead of k-shortest physical path mapping for the virtual links, k-shortest disjoint path pairs[2] mapping is used. Therefore, the VNO sees only a simple network, which is protected in the physical layer. The re-routing in case of a failure is realized in the physical layer by the corresponding PIP, i.e. the virtual topology remains unchanged and ideally the services are not disrupted. An example mapping using PIP-Resilience is shown in Figure 5.4b for a single service and $k = 2$.

Providing resilience for a certain network resource increases its price since PIP is offering an additional service. This price increase is called *Resilience Premium*, $r_{\text{PIP}}$. Therefore, for PIP-Resilience the cost minimization objective function is updated to include this additional cost component as shown in (5.13).

$$\min \quad \Big( \sum_{l \in L} r_{\text{PIP}} \left(\lambda_l \gamma_l + \theta_l u_l\right) + \sum_{v \in V} \left(\mu_v \alpha_v + \eta_v \omega_v\right) \Big) \tag{5.13}$$

## 5.4 Performance Evaluation of the Proposed Models

In this section, the performance evaluation of the proposed resilient virtual network design models is presented. First, the used simulation framework will be shortly introduced. Then, the proposed models will be compared with prior approaches in terms of applicability and virtual network complexity. Finally, a detailed analysis will be given to answer the question of at which layer it is better to provision resilience.

### 5.4.1 Simulation Framework

For the performance evaluation, we use two test networks as the physical network topology, namely the NobelUS and NobelEU [100] networks. For virtual network generation, first, we select a certain number of service nodes randomly from the physical network, where there is a uniform demand between all of them. Then, we solve the optimization problem for different resilience models, where each of them result in a different virtual network topology. They are then compared regarding their delay, cost, network resource usage and complexity performances until a confidence level of 95% and ±5% confidence interval is reached for the selected simulation aim value. Link diversity option is used for the simulations. However, our simulations show that node diversity option results in comparable service delay and virtual network setup cost values as link-diversity. Finally, the protection level $k$ is taken as 2 in the simulations as a practical value providing protection against single link and node failures, which are the most common physical failures [51]. The list of the parameter settings for this evaluation can be found in Section A.1.1. We vary the number of the service nodes to observe the effect of the network load and assume a uniform demand matrix, where there is a single unit demand between each service node pair. The assumption of a uniform demand matrix prevents the undesired effects like a biased emphasis on certain paths due to their high load.

In these simulations, we define six cost settings, which aim to yield an overview of all possible cost behaviors. These cost settings are defined in Table 5.1. They are presented as quadruples; {the fixed link setup cost, the capacity dependent link setup cost, the fixed node setup cost, the capacity dependent node setup cost}. These cases are chosen to investigate the effect of dominance and equality of the individual cost components in case of fixed and length-dependent cost factors. In the first two cases, the virtual link cost factors are taken as the physical length (in km) of the corresponding virtual link. In case of (L,L,A,A), the node cost factor is taken as 2000 for both physical topologies, which is an approximate value in the range of average virtual length link for the used test networks.

[2] K-shortest disjoint path pairs are the first $k$ disjoint path pairs when all the disjoint path pairs are listed in ascending order according to their total length.

| **Name** | **Setup cost of virtual links** | **Capacity dependent cost of virtual links** | **Setup cost of virtual nodes** | **Capacity dependent cost of virtual links** |
|---|---|---|---|---|
| (L,L,1,1) | Length | Length | 1 | 1 |
| (L,L,A,A) | Length | Length | Average | Average |
| (1,1,1,1) | 1 | 1 | 1 | 1 |
| (1,100,1,1) | 1 | 100 | 1 | 1 |
| (100,1,1,1) | 100 | 1 | 1 | 1 |
| (1,1,100,100) | 1 | 1 | 100 | 100 |

Table 5.1: Cost factors for the virtual network design models with connection services. ”Length” represents the physical length of a virtual link. ”Average” refers to the average shortest path length of a physical topology.

The first cost setting looks at the case of link cost dominance and second one offers results for a similar emphasis on both link and node costs.

Note that the length corresponds to the total physical length of the virtual link, i.e. in VNO-Resilience it is the length of the single physical path and in PIP-Resilience it is the sum of the lengths of the disjoint physical paths for each virtual link. Hence, the protected virtual links are in general more expensive than the unprotected ones. Similarly, for fixed link cost values we introduce a resilience cost premium $r_{\mathrm{PIP}}$ for PIP-Resilience. Its value is taken as 2 for the simulations. A detailed discussion on resilience premium is presented in Section 5.5.

The remaining cost settings assume a fixed link cost. (1,1,1,1) considers the case of equality of all cost components. Finally, in the last three cost settings, we investigate the effect of the dominance of the cost component with the weight 100, where the rest is kept minimum. 100 is chosen as an example value. In Section 5.5, we show that varying the exact values used in the cost settings does not affect the topological structure of the resulting virtual network.

### 5.4.2 Prior Approaches

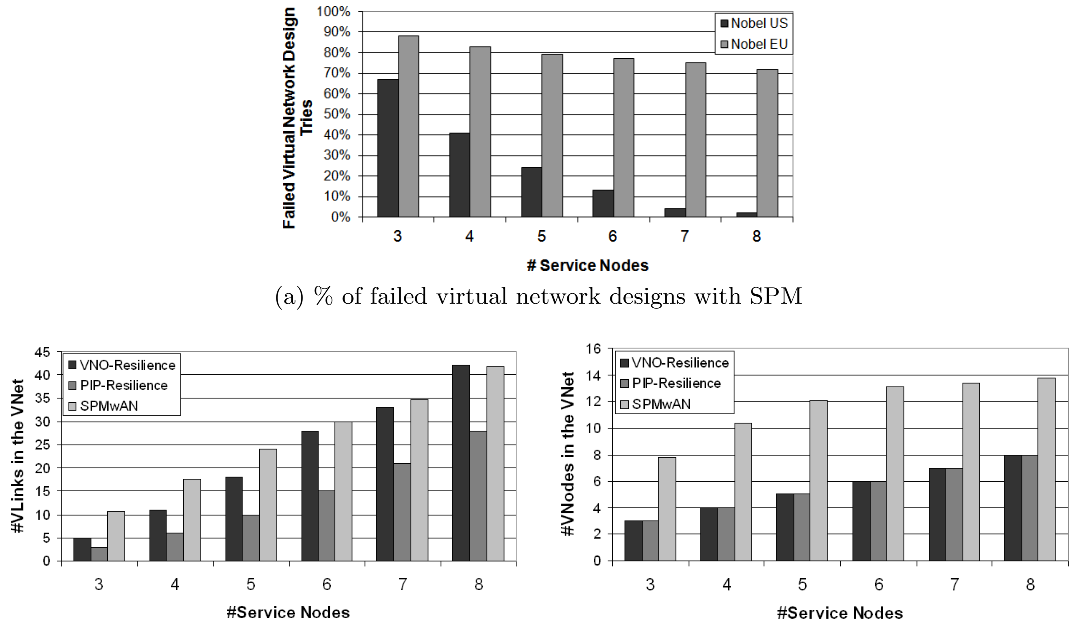
In this subsection, we present two prior approaches we consider in the performance evaluation of the proposed models. For both models, resilience is provisioned in the virtual layer.

#### 5.4.2.1 Shortest Path Mapping (SPM)

Shortest Path Mapping (SPM) model simplifies the virtual network design by limiting the virtual link mapping to the shortest path in the physical layer. This approach is adopted by certain literature [91, 92]. By eliminating the virtual network embedding optimization, the problem is simplified, however it becomes restricted in finding resilient virtual network solutions since physically disjoint virtual links might not be available due to the introduced limitation.

#### 5.4.2.2 Shortest Path Mapping with Additional Nodes (SPMwAN)

Shortest Path Mapping with Additional Nodes (SPMwAN) model is considered to relax the limitation in finding disjoint paths by keeping the shortest path mapping restriction for the virtual links. This is realized by extending the virtual node set, where the VNO can now use additional virtual nodes for routing purposes, which are neither the source nor the target node of any of the services. Similarly, the virtual link set is also extended to cover the possible links between all the node pairs in the new node set.



(a) % of failed virtual network designs with SPM

(b) Comparison of the number of virtual nodes with SPMwAN

(c) Comparison of the number of virtual links with SPMwAN

Figure 5.5: Virtual network design performance comparisons with prior approaches Shortest Path Mapping (SPM) and Shortest Path Mapping with Additional Nodes (SPMwAN)

### 5.4.3 Comparison with Prior Approaches

In SPM, direct shortest path mapping is used, and hence, it is not always possible to find physically disjoint paths to route the services and the resilient virtual network design cannot be performed. Figure 5.5a shows the ratio of the virtual network design tries with different service node sets, which failed to find a resilient solution during the simulations. Note that with increasing virtual network size, the probability to find a solution for SPM is increasing. However, for NobelEU network, even for virtual networks with 8 service nodes in over 70% of the tries, no solution could be found. Moreover, even if a solution is found for SPM, it always results in higher maximum delay compared with VNO-Resilience. This difference decreases with increasing virtual network size but is still over 20% for virtual networks with 8 service nodes on the test network NobelUS.

SPMwAN results in comparable latency values as the VNO-Resilience and overcomes the design restriction problem faced by SPM. However, firstly, it less scalable for larger physical networks. For our test networks, VNO-Resilience and PIP-Resilience simulations find a solution in a time interval of seconds but for SPMwAN, the simulation lasts for several minutes or even hours. Secondly, the resulting virtual network has more virtual links and nodes compared with the PIP-Resilience and VNO-Resilience cases as shown in Figures 5.5b and 5.5c, respectively. The virtual link numbers of SPMwAN and VNO-Resilience are closer for higher service node numbers. However, SPMwAN always has a higher number of virtual nodes independent of the virtual network size. Hence, especially for a high node cost factor, the network cost is drastically higher for SPMwAN. These results show that the proposed models outperform prior approaches both in terms of applicability and virtual network complexity and cost.

### 5.4.4 Comparison of VNO-Resilience vs. PIP-Resilience

In this subsection, a comparison of VNO-Resilience and PIP-Resilience is presented to answer the question of which advantages/disadvantages we observe when provisioning resilience at a certain layer. The metrics in question are virtual network setup cost, service latency, network resource usage and virtual network complexity. The first part deals with the comparison of cost and delay minimization and analyses the cost and delay metrics. Afterwards, the last two metrics are analyzed for each cost setting.

#### 5.4.4.1 Virtual Network Setup Cost and Service Latency

To evaluate the effect of different cost factors and optimization functions on the resulting cost and delay, we distinguish between seven cases as shown in Figure 5.6a. The first six cases, A-F, correspond to the six cost settings as A=(L,L,1,1), B=(L,L,A,A), C=(1,1,1,1), D=(1,100,1,1), E=(100,1,1,1) and F=(1,1,100,100), and for these cases cost minimization is used. Case G uses delay optimization. For this analysis the NobelUS network is used. Results for virtual networks with 3 service nodes are shown as a basis and practical example. The cost and delay differences shown in the figure are the relative differences of the two models, which are calculated by taking the difference of PIP-Resilience and VNO-Resilience values and dividing it by the VNO-Resilience value.

Cases B,C and F show that when the node cost is in the range of the link cost or higher, VNO-Resilience results in higher virtual network cost compared with PIP-Resilience. This effect is caused by the higher virtual node capacity usage in VNO-Resilience due to the two-paths routing inside the virtual network. In cases A and B, the cost of the link depends on the physical length of the link, and hence, cost optimization is aligned with delay optimization. In these cases, VNO-Resilience results in 20% lower delay than PIP-Resilience. When the delay optimization function is used as in case G, VNO-Resilience and PIP-Resilience result in comparable delay and cost values.

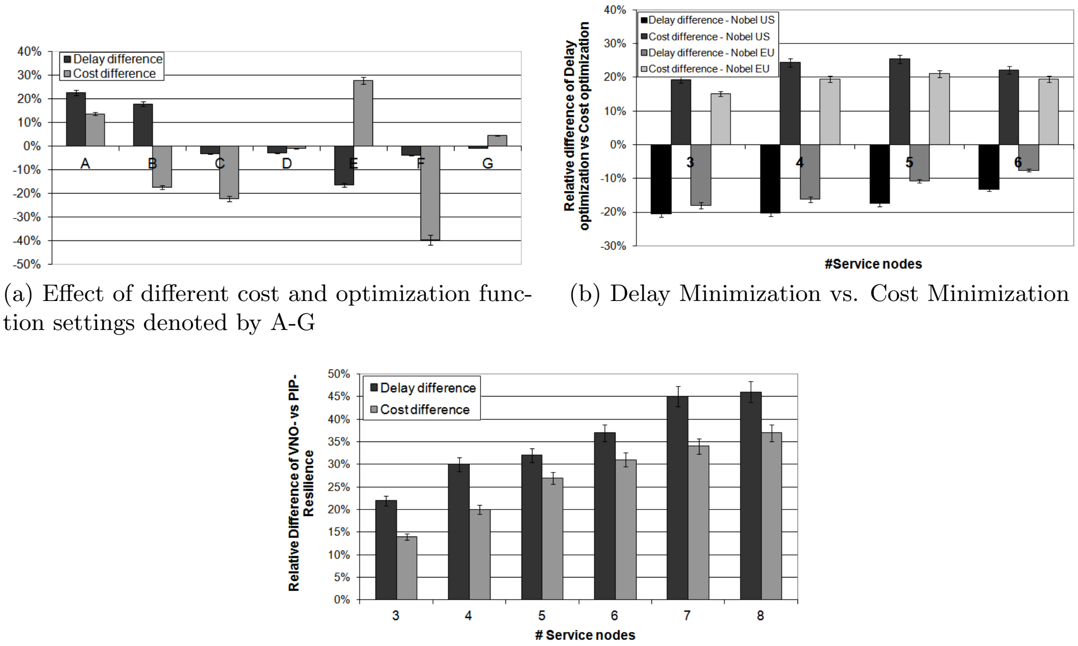
If we compare the results of delay optimization and cost optimization for VNO-Resilience using the cost setting (L,L,1,1) as shown in Figure 5.6b, it is observed that the delay minimization option results always in lower delay but higher virtual network cost. Increasing the number of the service nodes, decreases the delay difference of the two optimization functions but slightly increases the cost difference. Hence, the appropriate optimization function should be chosen according to both the number of service nodes and the cost factors.

In Figure 5.6c, cost minimization option and the cost setting (L,L,1,1) are used and delay and cost differences of PIP-Resilience and VNO-Resilience are presented. These differences increase with increasing virtual network size. As can be seen for larger virtual networks, a virtual network with PIP-Resilience costs on average 35% more than a virtual network with VNO-Resilience. Moreover, PIP-Resilience results in 45% higher service latency than VNO-Resilience for these settings. These results are obtained for the NobelUS network.

Finally, the results obtained using different test networks, namely NobelUS and NobelEU are compared. This comparison is given for the cost setting (L,L,1,1) with a varying number of service nodes in Figure 5.7. Comparing the cost and network resource requirement values for the two test topologies yields the conclusion that the trends remain the same for both topologies and the differences are emphasized with the NobelUS topology due to topological differences. This behavior applies for all the cost settings.

#### 5.4.4.2 Network Resource Usage

In this part, the amount of required network resources per virtual network is compared for VNO-Resilience and PIP-Resilience models. The amount of used bandwidth for a single physical edge is calculated by multiplying its length (in km) with the requested bandwidth on it. For a network wide calculation, this procedure is repeated for each physical edge. Recalling the fact that this comparison is more obvious for NobelUS topology, only selected results with this topology are presented. The preference for a resilience design in terms



(a) Effect of different cost and optimization function settings denoted by A-G

(b) Delay Minimization vs. Cost Minimization

(c) VNet Design for VNO-Resilience vs PIP-Resilience (cost optimization)

Figure 5.6: Virtual network design performance comparisons in terms of cost and delay for VNO-Resilience and PIP-Resilience

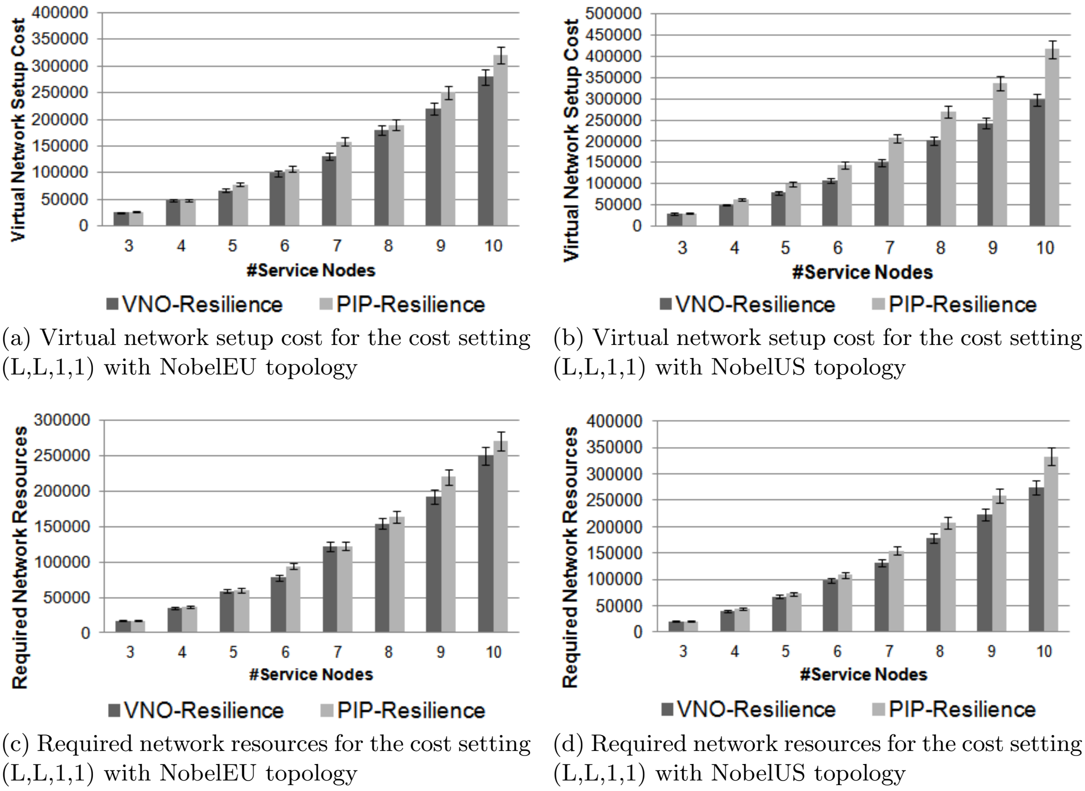
of network resource usage depends highly on the used cost setting. Cost setting (L,L,1,1) provides the most favorable results for VNO-Resilience with resource requirement reduction of 21% for 10 service nodes (45 services) compared with PIP-Resilience as shown in Figure 5.7d. Cost setting (L,L,A,A) is the second one where VNO-Resilience has a lower resource requirement per virtual network than PIP-Resilience by 9% as shown in Figure 5.8a. For the remaining cost settings, PIP-Resilience has a lower resource requirement. The case with (1,1,1,1) is shown in Figure 5.8b and the difference lies at 29%. For the cost settings (1,100,1,1), (100,1,1,1) and (1,1,100,100), PIP-Resilience has 34%, 7% and 27% lower resource requirement, respectively. This behavior is caused by the fact that making the link cost independent of its length causes the optimizer to choose arbitrarily long links and increases the network resource requirement especially for VNO-Resilience.

#### 5.4.4.3 Virtual Network Complexity

Recalling the results from Figures 5.5b and 5.5c, both PIP-Resilience and VNO-Resilience use the same number of virtual nodes since these are pre-defined by the set of service source nodes. However, VNO-Resilience results always in higher virtual link count due to the inclusion of redundant resources into the virtual network design. Even though this effect gets reduced with increasing virtual network size, it causes an increased complexity in terms of virtual network setup and maintenance.

#### 5.4.4.4 Performance Evaluation Summary

Summarizing the results for the different metrics, in terms of virtual network setup cost, the decision of which resilience model to apply depends on the selected cost setting. Dominance of link cost causes the virtual layer resilience to be more cost-efficient. For the case of cost equality or dominance of node cost, physical layer resilience should be preferred. In terms of service latency, VNO-Resilience is favorable in case the virtual link length is optimized with the cost, which is the case for the cost settings (L,L,1,1) and (L,L,A,A).



(a) Virtual network setup cost for the cost setting (L,L,1,1) with NobelEU topology

(b) Virtual network setup cost for the cost setting (L,L,1,1) with NobelUS topology

(c) Required network resources for the cost setting (L,L,1,1) with NobelEU topology

(d) Required network resources for the cost setting (L,L,1,1) with NobelUS topology

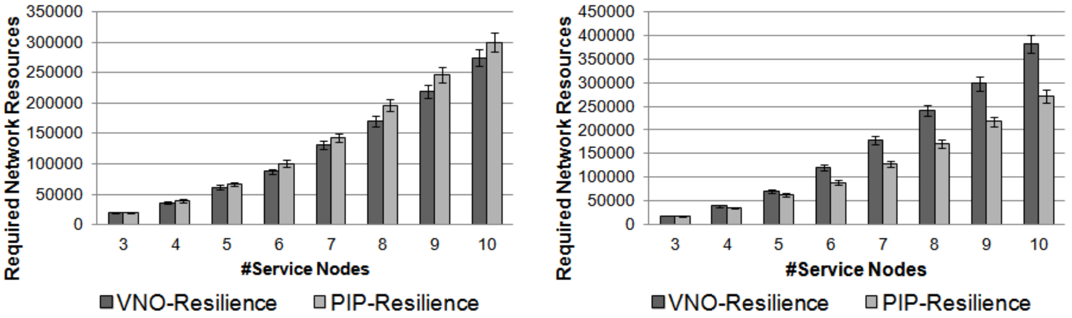
Figure 5.7: Virtual network setup cost and required network resources for the cost setting (L,L,1,1) with NobelEU and NobelUS topologies

Network resource requirement results are aligned with the latency results. Finally, due to redundancy provisioning within the virtual layer, VNO-Resilience always results in a higher number of virtual links and hence in higher virtual network setup and maintenance complexity.

## 5.5 Cost and Resilience Premium Analysis

Besides bringing more efficiency and flexibility into networks, network virtualization enables new business roles as introduced before. ISPs can be associated with different business roles, by either possessing physical resources, or by renting them from other providers. This raises open questions concerning the business relationships of these business roles. One of the open questions in this area is, if resilience is provisioned in the physical layer, how much fee should or can the provider ask for it in order to have a competitive pricing. In this section, we answer this question by an analytical analysis of different cost settings, thereby building a benchmark for future providers for assessing real world scenarios and deciding on their resilience pricing.

In the literature there are various works on the topics of network and resilience cost. These works differ mainly in their definition of cost. Some literature refer to cost as the price of installing and using physical network equipment [101, 97], where some refer to it as a performance metric like the delay or sparse capacity when optimizing for resilience [102]. The case of network virtualization differs from these cases, where we define cost as the price a VNO needs to pay to a PIP for the rental of certain virtual network resources. Since the physical network is already existing, and it is assumed that no additional dimensioning will be done for a new request, the virtual resource cost cannot be directly related to the installation cost. In such a scenario, determining the cost of resilience is hence also a



(a) Required network resources for the cost setting (L,L,A,A) with NobelUS topology

(b) Required network resources for the cost setting (L,L,1,1) with NobelUS topology

Figure 5.8: Required network resources for the cost settings (L,L,A,A) and (1,1,1,1) with NobelUS topology

white space. In general, a maximum resilience can be reached if unlimited resources are available. However, since this is not realistic, a cost-constrained network design has to be applied [103, 104]. The resilience design is as well not only dependent on the network cost but on the willingness of the client to pay [105]. Depending on the agreed service level agreements, a certain resilience and hence price level can be determined. A detailed analysis on this trade-off will be provided in Chapter 9. For now, our analysis will be based on the provisioning of 1:1 protection.

### 5.5.1 Cost Models

Providing virtual resources and establishing a virtual network compose a certain effort and cost for the PIP, and hence, a VNO needs to pay for this service. The rental price of a virtual resource can depend on various factors like its size, the service quality it can offer or the availability it can guarantee. The more properties a virtual resource has, the higher will be its price. However, how to decide on the amount of the premiums due to additional properties is an open business question.

Before going into the details of that question, we discuss the proposed cost model and introduce the generalized cost settings. In our cost model, we focus on the setup cost of the virtual network resources. This cost consists of two parts as it has been introduced in Section 5.2 in Figure 5.3. The first part is a fixed cost of setting up a new virtual machine or a new virtual link, and its value depends on the properties of these resources. In [106], a QoS differentiation model for virtual networks is introduced and in such a case the cost of each virtual resource depends on the QoS level it can offer. Other factors like the technology used in the underlying physical network resources, virtual resource's location or capabilities might also affect the fixed cost value. The second part is the capacity dependent cost, which relates the size of the requested virtual resource to its price. This cost function is linear where the granularity of the virtual resources is much lower or equal to one unit physical resource. For other cases, the non-linear capacity dependencies can be approximated by a linear function for lowly loaded networks. Therefore, the proposed model offers a basic tariff model for future providers.

We use six generalized cost setting options in this analysis as listed in Table 5.2, where each cost setting is defined as a quadruple ($\lambda_l$,$\theta_l$,$\mu_v$,$\eta_v$), presenting the fixed and capacity dependent costs of virtual links and nodes, respectively. These options define a set of cost setting varieties, where the dominance of all the cost components and the case of their equality are considered, to allow a representative assessment of real world scenarios. Their combinations are presented in the next section. The cost settings can be divided into

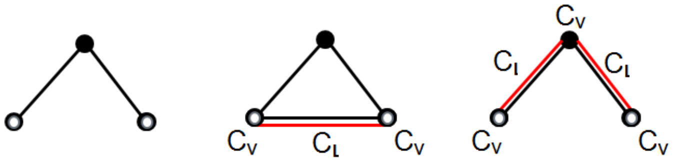
Table 5.2: Cost settings: marginal cases, $a_G$ is the average shortest path length in a physical topology $G$, $t_l$ is the physical length of a virtual link $l$ and the scaling value $x$ is a real number greater than 1.

| **Cost setting** | **Link Setup: Fixed Cost $\lambda_l$** | **Link Setup: Capacity Dependent Cost $\theta_l$** | **Node Setup: Fixed Cost $\mu_v$** | **Node Setup: Capacity Dependent Cost $\eta_v$** |
|---|---|---|---|---|
| (L,L,1,1) | $t_l$ | $t_l$ | 1 | 1 |
| (L,L,A,A) | $t_l$ | $t_l$ | $a_G$ | $a_G$ |
| $(1, 1, 1, 1)$ | 1 | 1 | 1 | 1 |
| $(1, x, 1, 1)$ | 1 | $x > 1$ | 1 | 1 |
| $(x, 1, 1, 1)$ | $x > 1$ | 1 | 1 | 1 |
| $(1, 1, x, x)$ | 1 | 1 | $x > 1$ | $x > 1$ |

two groups. In the first group, namely in the settings (L,L,1,1) and (L,L,A,A), the cost of a virtual link depends linearly on its physical length. In the former, the link cost is dominant, and in the latter, both link and node costs are in the same order of magnitude. The second option is having a length-independent value for the link cost, which can be determined according to the above mentioned factors. This is the case in the second group, where each individual link/node has the same fixed cost value. Depending on the investment and business model of the PIPs, the price ratio of the virtual links and nodes can vary. Therefore, we investigate all these cases to form a framework for future real world scenarios. We consider the cases of link and node costs to be in the same order of magnitude, capacity dependent cost for link setup to be the dominant cost, fixed cost of the link to be the dominant cost and finally the node cost to be the dominant cost in settings $(1, 1, 1, 1)$, $(1, x, 1, 1)$, $(x, 1, 1, 1)$ and $(1, 1, x, x)$, respectively, where $x$ is a real number greater than 1. Note that these are generalized cases of the cost settings used in Section 5.4 and therefore offer a validity check for them.

There is no need for an additional resilience premium if one of the first two settings are applied. In both cases, the link cost is dependent on the link's physical length, which is mapped on a single physical path in case of VNO-Resilience and on a disjoint physical path pair in case of PIP-Resilience. For the latter, the setup cost value depends on the sum of the lengths of the two paths, and hence, resilience premium is implicitly included to the cost of the resilient links. Therefore, for the resilience premium analysis only the cost settings with fixed cost values for link setup will be considered.

For PIP-Resilience, in case the link and node costs have the same value or if the capacity dependent cost for link setup or the node cost is dominant, the cost optimal virtual network with the introduced MILP model in Section 5.4 results always in a full-mesh network



(a) Existing VNet (b) Routing on a new single link (c) Routing on existing links

Figure 5.9: New service routing example for PIP-Resilience

topology. This behavior can be shortly explained via the example given in Figure 5.9. Assume that there is an existing virtual network as given in Figure 5.9a. If we want to route a new service between the marked nodes, we have two choices: either we can add a new link to connect these two nodes as shown in Figure 5.9b or the routing can be realized using the existing links, where the route of the new service can be two hops at minimum as shown in Figure 5.9c. For the former, the additional virtual network cost due to the new service routing, $\Delta_{cost1}$, is as given in (5.14) and for the latter, $\Delta_{cost2}$, is as given in (5.15), where $\lambda_l$ is the fixed cost and $\theta_l$ the capacity dependent cost for link setup, $\eta_v$ is the capacity dependent cost for node setup, $C_l$ is the bandwidth request of the new service on link $l$ and $C_v$ is the node resource request of the service on node $v$, which are assumed to be equal for all the used links and nodes, respectively.

$$\Delta_{cost1} = \lambda_l + C_l\theta_l + 2C_v\eta_v \tag{5.14}$$

$$\Delta_{cost2} = 2C_l\theta_l + 3C_v\eta_v \tag{5.15}$$

Given that $\lambda_l$ is negligible, $\Delta_{cost2}$ will be always greater than $\Delta_{cost1}$, and hence, routing each service on a single link connecting its endpoints is the most cost efficient solution, which results in a full-mesh topology for $(1, 1, 1, 1)$. Since increasing $C_l$, $C_v$, $\theta_l$ or $\eta_v$ increases the difference of $\Delta_{cost2}$ to $\Delta_{cost1}$, the settings $(1, x, 1, 1)$ and $(1, 1, x, x)$ also result in full-mesh topologies independent of the value of $x$. However, in setting $(x, 1, 1, 1)$, the aim is minimizing the number of the virtual links, and hence, the cost optimal virtual network results in a star topology, which is a connected topology with minimum number of links and lowest diameter [107]. By solving the PIP-Resilience model for varying values of $\lambda_l$, we have observed that this behavior occurs for $\lambda_l = x \geq 2$. If a resilience premium greater than 1 is used, the dominance of $\lambda_l$ can be already observed for slightly lower $x$ values.

For VNO-Resilience, analogous to PIP-Resilience, for the settings $(1, 1, 1, 1)$, $(1, x, 1, 1)$ and $(1, 1, x, x)$, the optimal virtual network obtained by solving the MILP has each service routing on a new link. However, since in this case resilience is provisioned in the virtual layer, each service needs to be routed on two physically-disjoint virtual paths inside the virtual network. Therefore, the cost optimal solution is connecting the endpoints of the service by two new links, which are physically disjoint. Therefore, in VNO-Resilience the cost optimal virtual network is a full-mesh multi-graph having always two parallel links between the node-pairs for all the three cost settings independent of the value of $x$. The setting $(x, 1, 1, 1)$ results for uniform demand matrices in a ring topology as the optimal solution, which is a 2-connected topology with minimum number of links. According to our observations using the VNO-Resilience model with varying $\lambda_l$ values, this behavior occurs for $x \geq 2$.

### 5.5.2 Resilience Premium Analysis

As introduced in Section 5.3, there are two principal ways of providing resilience in virtual networks, namely VNO-Resilience and PIP-Resilience. While realizing the virtual network design, there might be different reasons for a VNO to decide for one or the other. However, one important decision metric will be the cost of the virtual network. Therefore, PIPs which want to offer resilient virtual networks or resources should be in a position to have competitive offers compared with the case in which the VNO provisions resilience within the virtual layer. This can be realized by adjusting the resilience premium accordingly, and hence, it is very important for the future providers to have a benchmark to decide what they can afford while deciding for these values.

In this subsection, we discuss how the ratio of the link and node costs affect the selection of resilience premium and what is a feasible value for each of the different cases. This comparison is performed by calculating the results for VNO-Resilience and PIP-Resilience and determining the resilience premium values, which cause the virtual network cost with PIP-Resilience to be equal or less than with VNO-Resilience. This enables the physical layer resilience provisioning to be competitive in comparison to providing protection by the VNO.

The virtual network setup cost formula in case of PIP-Resilience is provided in (5.16), where $n_l$ is the number of virtual links in the virtual network, $C_{t,l}$ is the total requested link capacity, $n_v$ is the number of virtual nodes in the virtual network, $\mu_v$ is the fixed cost for node setup, $C_{t,v}$ is the total requested node capacity and $r_{\text{PIP}}$ is the resilience premium for having resilient virtual links.

$$\varepsilon_{\text{PIP-Resilience}} = n_l \lambda_l r_{\text{PIP}} + C_{t,l}\theta_l r_{\text{PIP}} + n_v \mu_v + C_{t,v}\eta_v \tag{5.16}$$

In case of VNO-Resilience, $r_{\text{PIP}}$ is omitted as given (5.17) because the PIP is providing non-resilient virtual links mapped on single physical paths, and resilience is provided in the virtual layer.

$$\varepsilon_{\text{VNO-Resilience}} = n_l \lambda_l + C_{t,l}\theta_l + n_v \mu_v + C_{t,v}\eta_v \tag{5.17}$$

As mentioned before, when the cost settings $(1,1,1,1)$, $(1,x,1,1)$ and $(1,1,x,x)$ are used, solving the virtual network design MILP results always in a virtual network having a full-mesh topology for PIP-Resilience and a full-mesh multi-graph topology with two parallel links for VNO-Resilience, independent of the value of the cost parameter $x$. Therefore, for these cost settings, the virtual network cost can be analytically calculated and it is given for PIP-Resilience and VNO-Resilience in (5.18) and (5.19), respectively, where $n_s$ is the number of the services. $C_t$ is the total demand request for all the services, where we assume a unit node capacity to be required for a unit link capacity. The actual amounts and types of link and node resources corresponding to their unit capacities should be defined according to the used technology and applications. The calculation is performed for a demand matrix having a service request between all virtual node pairs, where the services can have different demand amounts, which sum up to $C_t$.

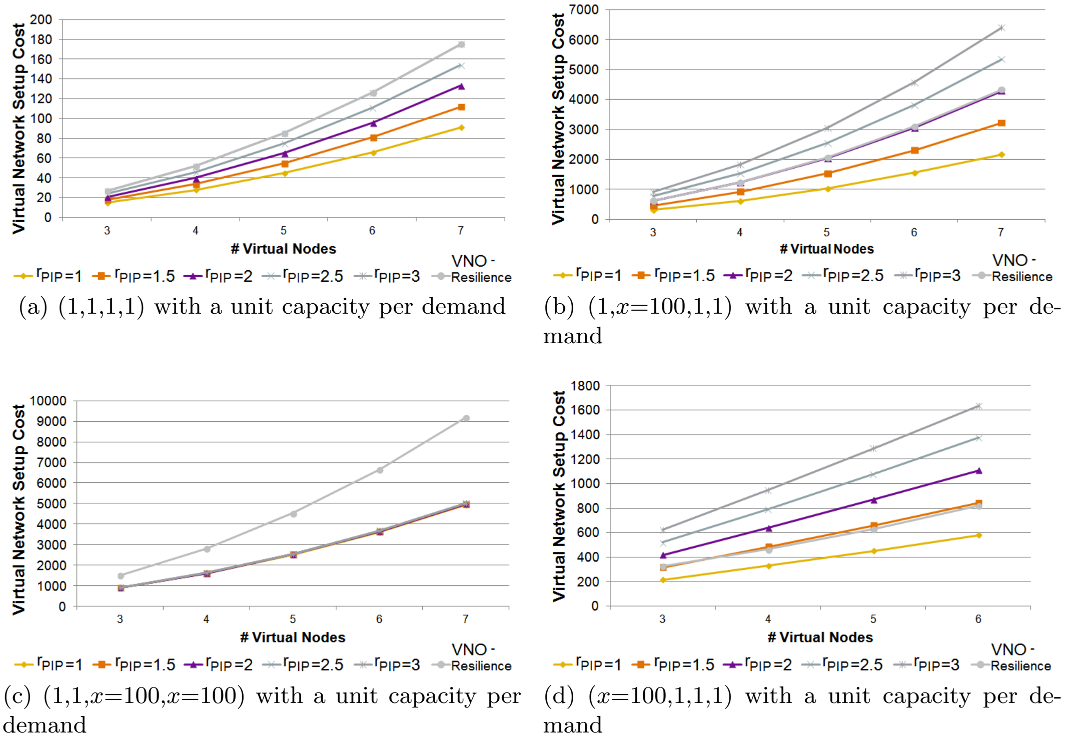
$$\varepsilon_{\text{PIP, full-mesh}} = n_s \lambda_l r_{\text{PIP}} + C_t \theta_l r_{\text{PIP}} + n_v \mu_v + 2C_t \eta_v \tag{5.18}$$

$$\varepsilon_{\text{VNO, full-mesh multi-graph}} = 2n_s \lambda_l + 2C_t \theta_l + n_v \mu_v + 4C_t \eta_v \tag{5.19}$$

As discussed before, if a PIP wants to offer resilience as a service to its customers, it should aim to have a competitive offer compared with the cost of VNO-Resilience. For the cost settings $(1,1,1,1)$, $(1,x,1,1)$ and $(1,1,x,x)$, this can be realized by having a resilience premium value equal to or less than the right-hand side of the formula given in (5.20). It shows that in case of full-mesh and full-mesh multi-graph solutions for PIP-Resilience and VNO-Resilience, respectively, an $r_{\text{PIP}}$ value of 2 can ensure PIP-Resilience to have always a lower virtual network cost compared with VNO-Resilience.

$$r_{\text{PIP}} \leq 2 + \frac{2C_t \eta_v}{n_s \lambda_l + C_t \theta_l} \tag{5.20}$$

In cost setting $(1,1,1,1)$, all the cost components are set to 1 unit. In this case, for $n_s << C_t$, the resilience premium for cost setting $(1,1,1,1)$ takes the value 4 and for



(a) (1,1,1,1) with a unit capacity per demand

(b) (1,$x$=100,1,1) with a unit capacity per demand

(c) (1,1,$x$=100,$x$=100) with a unit capacity per demand

(d) ($x$=100,1,1,1) with a unit capacity per demand

Figure 5.10: Comparison of VNO-Resilience with PIP-Resilience using different $r_{\mathrm{PIP}}$ values

$n_s >> C_t$, the value 2. Thus, the resilience premium for $(1, 1, 1, 1)$ should be selected from the interval $[2, 4]$ and the exact value can be calculated using (5.20) by inserting the actual number of services and the requested capacity. An example with one unit capacity per demand is shown in Figure 5.10a, where the equality of the two models is reached by $r_{\mathrm{PIP}} = 3$, i.e. with $r_{\mathrm{PIP}} \leq 3$ a PIP can have a competitive offer compared with the VNO-Resilience.

For cost setting $(1, x, 1, 1)$, all the cost components except $\theta_l$ have a negligible value and $\theta_l$ has the value $x$, which is greater than 1. In our models, resilience is provided only for the links, and hence, resilience premium is applied only to the link cost. Therefore, with an increasing value of $x$, the effect of the resilience premium is emphasized and as $x$ goes to infinity, the limit value of the resilience premium for cost setting $(1, x, 1, 1)$ reaches 2. Therefore, it is advisable to take the value of the resilience premium as 2, which ensures a competitive offer for all possible cost values and the exact limit can be calculated using (5.20). The example with $x = 100$ is shown in Figure 5.10b, where PIP-Resilience with $r_{\mathrm{PIP}} = 2$ is just under the curve of VNO-Resilience.

The last cost setting, where we observe a similar behavior as in $(1, 1, 1, 1)$ and $(1, x, 1, 1)$, is the cost setting $(1, 1, x, x)$, with the difference of the dominance of the node cost. Since the virtual nodes included in the virtual network are only the service nodes, which are predefined, the fixed cost for node setup does not affect the design of and routing in the virtual network, and hence, it does not affect the value of the resilience premium either, which can be seen in (5.20). Therefore, we do not differentiate the dominance of the two node cost types unlike for the virtual links.

Also for cost setting $(1, 1, x, x)$, a resilience premium $r_{\mathrm{PIP}}$ of value 2 will always ensure PIP-Resilience to have a lower cost compared with VNO-Resilience. Increasing the value $x$ makes the price advantage of the PIP almost independent of the resilience premium and enables the PIP to choose the $r_{\mathrm{PIP}}$ value freely. This effect is reasonable since resilience is only provided for the virtual links and for this cost settings the node cost is the dominant component. Finally, for a fixed and relatively large $x$ value, changes in the $r_{\text{PIP}}$ value do not affect the results as shown Figure 5.10c.

As mentioned before, using cost setting $(x, 1, 1, 1)$, the cost optimal virtual network has a star topology in PIP-Resilience and a ring topology in VNO-Resilience for $x \geq 2$ and uniform demand matrix. The virtual network setup cost for this cost setting is given in (5.21) and (5.22) for PIP-Resilience and VNO-Resilience, respectively.

$$\begin{aligned} \varepsilon_{\text{PIP},star} = (n_v - 1)\lambda_l r_{\text{PIP}} + & \frac{2(n_v - 1)}{n_v} C_t \theta_l r_{\text{PIP}} + n_v \mu_v \\ & + \left( \frac{2(n_v - 1)}{n_v} + 1 \right) C_t \end{aligned} \tag{5.21}$$

$$\varepsilon_{\text{VNO},ring} = n_v \lambda_l + n_v C_t \theta_l + n_v \mu_v + (n_v C_t + 2C_t)\eta_v \tag{5.22}$$

In this cost setting, the fixed cost of link setup, $\lambda_l$, is the dominant cost component. Resilience is provided solely for the links, and hence, resilience premium is applied only for the link cost. Therefore, the dominance of link cost limits the value of the resilience premium similar to cost setting $(1, x, 1, 1)$. Moreover, since in case of $(x, 1, 1, 1)$ the cost optimal virtual network has a star topology in PIP-Resilience ($n_v - 1$ links) and a ring topology in VNO-Resilience ($n_v$ links), the difference in the number of the virtual links of the two topologies is 1. For large values of $x$, the resilience premium is limited to the ratio of these values as given in (5.23). The example with $x = 100$ is shown in Figure 5.10d, where the limiting $r_{\text{PIP}}$ value lies around 1.5.

$$r_{\text{PIP}} < r_{\text{PIP},(x,1,1,1)} \approx \frac{n_v}{n_v - 1} \quad \text{for large } x \tag{5.23}$$

For virtual networks with a large number of nodes, resilience premium tends to go to 1, which means that the PIP will not be able to charge for the extra resilience service it is providing to the VNO. Therefore, for this cost setting it is advisable for the future providers to favor resilience provisioning within the virtual layer.

Finally, we analyze the combinations of the given cost settings. For the case of dominant capacity dependent link cost and node cost, namely $(1, x_1, x_2, x_2)$, the equations (5.18)-(5.20) apply and the higher the node cost, the more independent is the choice of the $r_{\text{PIP}}$ value from the virtual network cost.

When both link cost components $\lambda_l$ and $\theta_l$ have higher values and node cost is negligible, $(x_1, x_2, 1, 1)$, the behavior of the model depends on these values. For $\lambda_l = \theta_l >> \eta_v$, (5.20) applies, and $r_{\text{PIP}}$ value goes to 2. For $\lambda_l > \theta_l$ the optimal virtual network solutions start to turn into star and ring topologies for PIP-Resilience and VNO-Resilience, respectively. Therefore, $r_{\text{PIP}}$ needs to be reduced, and hence, it is advisable to prefer resilience provisioning in the virtual layer.

Finally, if the fixed cost of the links and the node cost are dominant, namely for $(x_1, 1, x_2, x_2)$, the cost and topological behavior again depends on the values of these components. For $\frac{\eta_v}{r_{\text{PIP}}} < \lambda_l \leq \eta_v$, PIP-Resilience results in a star topology but VNO-Resilience in a full-mesh multi-graph. In this case, $r_{\text{PIP}}$ value depends on the ratio of the number of the virtual links, which is directly related to the number of the virtual nodes, as given in (5.24). For $\lambda_l > \eta_v$, VNO-Resilience also tends to have a ring topology, and hence, $r_{\text{PIP}}$ goes to (5.23).

$$r_{\text{PIP},(x_1,1,x_2,x_2)} \leq n_v + \left( \frac{n_v}{2} + 1 \right) \frac{\eta_v}{\lambda_l} \tag{5.24}$$

### 5.5.3 Summary

Resilience will continue to be a key issue in future networks. Where to provision resilience in this architecture is an open question. One of the metrics for the VNO's decision of either providing it itself within the virtual network or delegating it to the PIP will be the cost of this service. Therefore, the PIPs require a good strategy when determining their resilience premium. We present a detailed framework, which can be used as a benchmark when selecting the resilience premium value. We provide the limits of the resilience premium for all different cost settings, which make PIP-Resilience competitive in terms of the virtual network price. We show that a resilience premium of value 2 always ensures PIP-Resilience to have a lower cost compared with VNO-Resilience except for the dominance of the fixed cost of setting up a new virtual link. For this case, providing resilience in the physical layer is not feasible in terms of pricing. Finally, the higher the node cost, the freer are the PIPs to determine the resilience premium independent of virtual network cost considerations.

## 5.6 Heuristic Algorithms for Resilient Virtual Network Design

In this section a general heuristic framework for virtual network design will be introduced. It will be shown which modifications are necessary compared with the traditional heuristics in order to be able to apply them to virtual networks. Finally, two selected heuristics will be shortly presented, whose implementation details will be shown later in Chapter 7 using shared protection in virtual networks as a use case example.

In the literature there is extensive work available on routing heuristics [108, 109, 110, 111, 112, 113, 114, 69, 115, 116]. They can be divided into several groups according to their optimization framework. Some of them perform the optimization network-wide by taking into account a bundle of services [109, 110, 111]. The algorithm in [109] works on the demands sequentially, where these are ordered in descending bandwidth request amounts. Cinkler et al. [111] compare the approaches of having sequential and parallel processing of the set of demands and they also use a method of having an additional loop for improving the obtained results. Some literature, however, defines the objective functions only at the level of each node-to-node traffic flow [108, 112, 113]. These papers generally focus on solving a special problem like shared protection [108], QoS constrained routing [113] or minimum-cost multicast routing with bounded-delay [112].

In this thesis, we will utilize network-wide optimization. In this case, the objective function is defined to realize network-level optimization rather than single demand-level optimization. We use as the objective function virtual network setup cost minimization, namely the same as for the MILP as given in (5.1). Cost of an resource in the sense of heuristics can be defined in various ways. It might be for example the length of a link or it might be assigned dynamically depending on the current load of the link to reach load balancing within the network [2]. In our case we define the cost as the price a VNO needs to pay to a PIP to acquire this virtual resource. The main difference of our case with the traditional routing heuristics is that in a virtual network the use of each additional link and node causes an extra fee. Therefore, there is a need for favoring the re-use of existing resources. This can be done by adopting the routing logic from shared protection heuristics [117, 118, 119], where the already used links' cost is minimized, and hence, the algorithm tends to prefer such links. In our case, we use the fixed and capacity dependent cost components for the links/nodes, where the fixed cost of a link/node is set to 0 if that link/node is already used by a former demand. In that case, the additional cost due to the current demand is only the capacity dependent cost, which is calculated as the unit capacity cost of the link/node multiplied by the corresponding capacity requirement of the demand. For a new link/node setup, the fixed setup cost has to be considered as well. The general basic structure of the virtual network design heuristics is given in the following.

1. **Sort the demands**: This part is optional and can be used if an a priori knowledge about a good sorting exists. However, in general certain orderings (e.g. according to requested bandwidth or path length) offer better solutions than random sorting.

2. **Initial routing and virtual resource selection**: The input at this step is the physical network topology, the virtual link and node candidates and the ordered set of demands with their requirements.

   ```
   Initialize link and node costs
   for each demand $d$ do
     Do minimum cost routing in the virtual layer (implicitly includes virtual link/node selection) according to the requirements of the demand (bandwidth/node resource requirement, resilience option, QoS option, etc.)
     Update the link/node cost values
   end for
   Calculate the virtual network setup cost
   ```

3. **Iteration**: Once the initial routing and virtual resource selection is performed, the algorithm has calculated a feasible solution comprising the service routing and virtual network design. This solution can be potentially improved by using the following iteration.

   ```
   while The routing of at least one of the demands changes and the maximum number of iterations is not reached do
     for each demand $d$ do
       Delete the initial routing of the demand $d$
       Re-calculate the updated link and node costs due to removal of $d$
       Re-route the demand $d$
       Calculate the current value of the objective function
       Keep the new routing if the acceptance condition is met, otherwise re-do the old routing
     end for
     Break if the stopping condition is met
   end while
   ```

The algorithm structure is composed of three parts, namely the sorting of the demands, the initial routing and the iteration step aiming to improve the initial routing and virtual network topology design solution. The iteration runs until either the routing of all the demands remains unchanged in one iteration step or the maximum number of iterations is reached. The routing of a demand is changed only if the new routing reduces the overall virtual network setup cost. This approach is called the *HillClimber* algorithm.

A second option, the *kBest* algorithm, has the same general structure, with the difference that instead of calculating a single solution and saving it for each demand at each step, $k$ solutions are kept. Each of these solutions is the input to the routing algorithm of the next demand, resulting in $k^2$ routings, from which the best $k$ are kept comprising a tree of solutions. At the end, the best overall branch of the tree is returned. For the iteration part, only the first service routing is alternated keeping the others unchanged and if the solution is improved, the other demands are re-routed following the same algorithm again. This approach offers more choices in each iteration step but reduces the flexibility of the general re-routing.

More details on these two algorithms are given in Chapter 7, where both algorithms are implemented to solve the shared protection problem in virtual networks. Shared protection is selected as the case study to evaluate the performance of the proposed algorithms due to two reasons. Firstly, it is an interesting problem lowering the virtual network cost for a VNO and improving the physical network utilization for a PIP, and hence, creating a win-win situation in a virtual network environment. And secondly, due to the increased complexity of the problem, optimization models prove not be scalable, creating the need for efficient heuristic methods.

## 5.7 Summary

Network virtualization is seen as a key enabler for future networks. It allows more flexibility, efficiency and service-tailored design compared with traditional network architectures. In the network virtualization architecture, where different players, namely the PIPs, VNOs and SPs realize different tasks and have a complex business relationship, how to design resilient virtual networks is an open question. This chapter answers this question by introducing optimization models for resilient virtual network design, where resilience can be provided either in the physical or in the virtual layer. We then present the performance evaluation of the proposed models by comparing them with prior approaches as well as analyzing the effect of having resilience in different layers. We show that the proposed models outperform the prior approaches by applicability and virtual network setup cost and complexity. Resilience provisioning at the virtual and physical layers is evaluated in terms of virtual network setup cost, service latency, physical network utilization and virtual network complexity. For the virtual network setup cost evaluation we have designed six different cost settings, which enable to analyze the effect of the dominance of different cost components like the virtual link or node cost. The simulation results are generated using selected values. Afterwards, we investigate the behavior of the cost settings with varying numerical values and show that the behavior is independent of the selected values. Moreover, we provide a framework about choosing the value of the resilience premium, the additional price a VNO needs to pay to a PIP to receive resilience service from the physical layer. Finally, we introduce a general framework for constructing heuristic algorithms for resilient virtual network design.

The following research questions from Section 1.3 are answered in this chapter:

**Q1.1:** Does the prior art provide answers to the resilient virtual network design problem? If not, where are the shortcomings?

In this chapter, we provide a literature survey related to virtual network design. We group the related work into three parts. The literature in the area of resilience in multi-layer networks lacks the notion of virtualization and thus the virtual network architecture with different business roles and their relationships. The works in the area of overlay networks assume that the virtual network topology is given and is already mapped onto the physical substrate. Similarly, literature in the area of VPNs and virtual network embedding assume also the virtual network topology to be given together with the service routing and try to embed it onto the physical network. Resilient virtual network design, however, can be enabled by optimizing the service routing and virtual network mapping simultaneously. Therefore, both of these literature directions provide sub-optimal solutions and due to their assumption on the existing virtual network topology, they fail to answer the question of how to design resilient virtual networks.

**Q1.2:** How can the design of resilient virtual networks be performed at the VNO layer using the input coming from the infrastructure providers, PIPs, and their customers, SPs?

We answer this question by proposing novel resilient virtual network design models formulated as MILPs. The introduced models take as input the available resources from the

PIP layer and the connection service requests from the SPs. The models allow simultaneous optimization of the service routing and virtual network mapping, hence preventing sub-optimality. We show that the proposed models outperform the prior approaches in terms of applicability and virtual network setup cost and complexity.

**Q1.5:** To cope with the possible scalability problems of the virtual network design models, what kind of heuristics can be used for resilient virtual network design?

We introduce a general framework for building heuristic algorithms performing virtual network design. The general idea is based on the heuristics in the area of shared protection, where the re-use of certain links or paths are favored to decrease the redundant capacity usage. In the case of virtual networks, the re-use of already used virtual links and nodes is preferable to lower the virtual resource setup costs. Based on this general logic, two algorithms are shortly introduced, namely the HillClimber and kBest algorithms, whose implementation and performance evaluation are presented in detail in Chapter 7.

**Q2.2:** Does virtual layer resilience bring any benefits in terms of virtual network setup cost, service latency, physical resource utilization and complexity?

In terms of virtual network setup cost, the preferred resilience model depends on the used cost setting. Dominance of link cost causes the virtual layer resilience to be more cost-efficient. For the case of cost equality or dominance of node cost, physical layer resilience should be favored. Virtual layer resilience lowers the service latency in cases where the virtual link length is optimized together with the cost. Similarly, it reduces the network resource requirement for the same service set, where these results are aligned with the latency results. Finally, due to redundancy provisioning within the virtual layer, virtual layer resilience always results in a higher number of virtual links, and hence, increasing the virtual network setup and maintenance complexity.

## 5.8 Statement on Author's Contributions

The Sections 5.1 - 5.4 are an extended version of [88], where the proposed models and performed evaluations presented in this publication have been carried out by the author. In the thesis a more elaborated comparison with related work is provided in Section 5.1, a detailed explanation of the used inputs and the outputs of the MILP is presented, and the cost model used in the objective function is described in higher detail. Moreover, cost calculation constraints for the latency minimization case are added, which enable better cost results when latency is optimized. Therefore, Figure 5.6b is updated accordingly. Finally, network utilization results are added to the performance evaluation section. In addition to the work presented in [88] a cost and resilience premium analysis and a heuristic framework are provided.

# 6. Combined Optimization of Networks and Clouds for Virtual Network Design

Businesses and applications are increasingly based on cloud technologies, wherein infrastructure, software and platform as a service are important types of services. Therefore, assurance of end-to-end Quality of Experience (QoE) for cloud services is of high importance, in particular for business-critical applications. According to a worldwide survey of over 3700 companies conducted in 2011, businesses adopting cloud services are primarily concerned with reliability, while performance ranks third in the list of concerns [3]. Therefore, cloud providers offer solutions to address these concerns. However, such solutions are focused on performance and connectivity within the cloud and only insufficiently address communication networks, which are an important cause of unacceptable latencies and service outages. Since communication networks and cloud domains are typically operated by different entities, it is difficult at the moment to offer end-to-end QoE guarantees for cloud services. Furthermore, services are typically provided by a single cloud provider. In the event of a complete DC failure, the recovery of services may cause long outages depending on the geographical diversity and availability of the cloud provider's resources. As discussed in the previous chapters, the concept of network virtualization with combined control of network and IT resources and with migration possibilities offers a promising solution for these problems. This enables a complete overview of the available virtual resources of various physical domains and an optimized operation of cloud networks.

In this chapter, we generalize definition of the virtual network resources to include cloud resources, and hence, they comprise virtual network links and nodes and IT resources like compute and storage resources as it has been shown in Figure 1.2. As introduced in the previous chapters, in a network virtualization environment, new business roles can be established, which realize different tasks and trade virtual resources between them [120, 38, 36]. Therefore, new control mechanisms and interfaces are necessary to realize the setup and operation of these heterogeneous virtual networks. There are already several suggestions in the literature for possible realizations of combined control of IT and network resources using virtualization [121, 122]. There are also some commercial offers from e.g. Amazon [123], where a virtual network is deployed for the connectivity to the cloud although it still lacks resilience and end-to-end QoE guarantees. Therefore, as a solution, we propose in this chapter novel virtual network design models to enable optimal provisioning of cloud services with end-to-end availability and latency guarantees, which provision resilience in the virtual layer, in the physical layer or using a combination of both.

Currently, how to design such virtual networks enabling end-to-end resilience for cloud services over networks and at which layer to apply resilience are open research questions. Our contributions provide solutions for both of these issues. In this chapter, first, an initial analysis is provided to show how much benefit virtual layer resilience offers in terms of latency by evaluating the different cloud connection scenarios for existing virtual networks. Afterwards, we introduce the optimization models for end-to-end resilient virtual network design for cloud services. These models cover different options for resilience provisioning, namely at the virtual layer, at the physical layer and the hybrid models using a combination of these two approaches. We provide analytical delay and cost analysis for these models and present a general framework for heuristics to be used in resilient virtual network design for cloud services. The chapter is concluded with a summary of the main contributions.

This chapter is based on three publications. Section 6.2.1 is based on [83], Section 6.2.2 is based on part of [88] and Sections 6.3 and 6.4 are an extended version of [124] and [125].

## 6.1 Related Work and Contributions

In a virtual network environment, a VNO will receive the cloud service requests from its customer SPs and will have an overview of the available virtual resources of the PIPs as they are advertised to it. In the light of this input, it needs to design a cost-efficient virtual network satisfying the service requests. In this section, we analyze the existing literature in terms of the solutions for building resilient virtual networks and also in terms of the solutions offered for the anycast routing problem.

The existing literature about virtual network design is twofold as it has been discussed in Chapter 5. The first type of related work is in the area of resilient overlay networks [126, 127]. In [126] IP/MPLS overlay network design over WDM networks is presented, where in the design of LSPs Shared Risk Link Group (SRLG) diversity is taken into account. The work in [127] also deals with the design of overlay networks, where it additionally allows the choice of the overlay nodes to be installed in different locations in the physical network. However, in both of these works a fixed mapping of the virtual links onto the physical layer is assumed, and therefore, only service routing using these virtual links is optimized. However, a virtual link can be mapped onto the physical substrate in many different ways. Considering these options rather than assuming a fixed mapping allows better optimization in virtual network design.

The second type of literature is in the area of resilient multi-layer networks [2] and virtual network embedding [128, 129, 130, 131]. The work in [128] states that the connectivity metrics have different signification in single-layer and multi-layer networks and proposes survivable lightpath routing algorithms in the light of new connectivity metrics. The virtual layer topology is given as an input and the lightpath corresponding to each virtual link is then routed in the physical layer. Dietrich et al. [130] propose the usage of demand matrices instead of providing the request as a virtual network topology since the latter is restricting the design. However, they only focus on virtual node mapping and consider only the physical layer routing between these nodes. The papers [129] and [131] deal with virtual network embedding problems including IT infrastructures. However, two weaknesses are observed in their approach. Firstly, they only focus on communication between servers/facility nodes. This approach is useful for intra-DC scenarios but for inter-DC communication or communication between a service source node and a DC site this approach is not directly applicable. Moreover, like the other virtual network embedding literature, they assume the service request to be defined as a virtual network topology, i.e. they assume that the virtual network topology is given as an input. This assumption causes a big limitation in terms of overall optimization and it is not applicable to the cases, where the virtual network topology is unknown and has to be designed according to the available virtualized physical resources and service requests of the SPs, which is the case in network virtualization.

Finally, the literature available for optimal server selection and routing of anycast services in the physical layer for intra-DC and inter-DC networks, [132, 133, 134, 135], lacks the treatment of the resilient network design in the virtual layer. Therefore, even though this type of literature provides the basics for anycast routing scenarios, it is not applicable to the problem space of designing resilient virtual networks for cloud services.

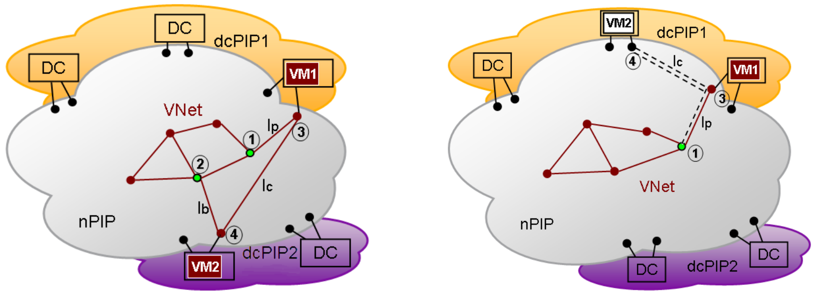
In a virtual network environment, there are service requests, which are routed within the virtual network and the virtual network is mapped onto the physical substrate. We have seen that the existing literature either assumes one of the service routing or virtual network mapping to be fixed or discards totally the virtualization aspects of the problem. In this chapter, we propose novel virtual network design models for cloud services, which overcome these problems by applying simultaneous service routing and virtual network mapping optimization. We show that the proposed models outperform prior approaches via extensive simulations. We also evaluate if virtual network resilience for cloud services offers any benefit in terms of virtual network setup cost, network resource usage, service latency and virtual network complexity. To answer this question we first present and analyze an initial cloud connection scenario for existing virtual networks and then evaluate the performance of the proposed virtual network design models with end-to-end optimization of network and IT domains. An analytical cost and latency analysis and a general framework for virtual network design heuristics for cloud services are also presented.

## 6.2 A First Analysis of Cloud Connection Models

There are two fundamental alternatives for providing resilience to cloud services in a network virtualization architecture, namely provisioning it in the virtual layer by the VNO or in the physical layer by the PIP(s). Before going into the details of the resilient virtual network design for cloud services, in this section, we present a first analysis to investigate how much gain the virtual layer resilience can offer under various circumstances. The first subsection assumes a given virtual network and establishes resilient cloud connections to this network once having virtual layer and once physical layer resilience and then compares the performance of these two solutions. In the second subsection, however, the virtual networks of the two cases are designed according to the resilience models in Chapter 5 and are hence different for each case. Afterwards, these two networks are connected to the cloud according to their selected resilience option realizing virtual or physical layer resilience for both the network and the DC resources. The performance evaluation of the end-to-end resilience designs for cloud services are presented at the end of this section and compared with the models providing only DC and connection path resilience.

### 6.2.1 Connecting Existing Random Virtual Networks to Clouds

In this section, we present the details of the two resilience designs, where only resilience against physical DC failures and failures in their connection links with the virtual network are considered. In both cases, we use the same scenario where resilient cloud connections are requested for an existing virtual network, and the virtual network is extended to satisfy the given cloud connection requests according to the selected resilience design. In other words, with the arrival of the cloud connection requests, the virtual network modeled as the graph $G_l(V, L)$, with $V$ being the set of the virtual nodes and $L$ the set of the virtual links, is extended to enable cloud connections. The new graph is given as $G'_l(V', L')$. These



(a) VNO designs a resilient cloud connection to protect against DC failures. In normal operation, VM1 and the link $l_p$ are used and in case of failure, the services are routed over the link $l_b$ to VM2.

(b) The cloud connection in the virtual layer is designed without any protection against DC failures. In case of failure, the services are redirected to VM2 over the path $l_c$ in the physical network transparently to the VNO.

Figure 6.1: DC connection models with resilience (a) in the virtual and (b) in the physical layer

graphs are mapped onto the physical network $G(N, E)$, where $N$ and $E$ are the set of the physical nodes and links, respectively.

#### 6.2.1.1 VNO-Resilience for Cloud Connections of Existing Virtual Networks

In VNO-Resilience, cloud connections are designed to provide resilience without needing any further recovery action from the PIPs. The VNO just requires the location information of the DCs, at least in terms of availability regions, in order to provide IT service survivability. At the time of the cloud connection request, the available network and DC resources from several nPIPs and dcPIPs are advertised to the VNO. It is assumed that all the services running in this virtual network will be served by one primary DC and (at least) one Disaster Recovery (DR) site. When a failure is detected at the primary site, the services are routed in the virtual layer to the DR site. VNO optimizes its selection for the servers, where each server is assumed to have the same computational properties. The selection of the servers is performed by calculating the routes for all the service requests to each server and selecting the two, which provide the lowest maximum or average latency values for all the services. In our latency calculations only the propagation delay is considered, which is calculated for the average and maximum latency cases as given in (6.1) and (6.2), respectively.

$$d_{\text{avg}} = \frac{1}{|S|} \sum_{s \in S} \sum_{e \in p_s} \frac{l_e}{0.67c} \tag{6.1}$$

$$d_{\max} = \max_{s \in S} \sum_{e \in p_s} \frac{l_e}{0.67c} \tag{6.2}$$

In (6.1) and (6.2), $S$ is the set of the requested cloud services and $c$ is the speed of light. Note that in these calculations transmission in fiber is assumed, where the light travels with a speed of $0.67c$. For the calculation of the propagation delay of a service $s \in S$, the service is first routed in the virtual network over the shortest path $p_s$ to the candidate DC. The multiplication of the sum of the lengths $l_e$ of the physical links $e \in E$, on which $p_s$ is mapped, with the speed of light in fiber gives the propagation delay for $s$.

Figure 6.1a shows the resilience design for VNO-Resilience. In normal operation, the services are routed over the connection path of the virtual network to the primary DC hosting VM1, namely $l_p$. In case there is a failure in the primary DC, the services will be routed in the virtual network to the DR site hosting VM2 using the backup link $l_b$. The link between the two DCs, $l_c$, is established for synchronization and data migration purposes. Note that it is allowed to choose the two DCs from different dcPIP networks. By the choice of the DCs, e.g., the distance to the virtual network, the performance or cost can play a role. In our scenario only the distance to the virtual network is selected as the decision parameter.

The VNO-Resilience model becomes non-scalable with increasing number of DCs due to the large number of possible DC-connection node combinations. Therefore, we introduce a heuristic, where the primary DC and its connection node, node 1, are chosen first according to the maximum end-to-end delay it provides. However, the path $l_p$ is not fixed but rather a candidate path list is created holding the k-shortest paths between the primary DC and node 1. Afterwards, the DR site and its connection node, node 2, are chosen to minimize the end-to-end delay considering both the virtual network delay and the routing on $l_p$, $l_b$ and $l_c$, where all of these links are mutually physically disjoint. The end-to-end delay performance difference of the optimal case and the heuristic remain in $\pm 5\%$ interval for the NobelUS network with different DC and dcPIP settings and it is hence negligible.

#### 6.2.1.2 PIP-Resilience for Cloud Connections of Existing Virtual Networks

In PIP-Resilience, the virtual network is physically connected to two DC sites where from the VNO point of view it is observed as a single resilient connection. In case of a failure at the primary site, the traffic is re-directed to the DR site in the physical network. The primary DC is first chosen by the VNO via latency optimization same as in the VNO-Resilience case. The dcPIP owning this DC is responsible for the resilience of the cloud services. Thus, the DR site can only be chosen from the domain of this dcPIP. The choice of the DR site then depends on the internal strategy of this dcPIP and possibly on the contract with the VNO. Two possible strategies for the choice of the DR site by the dcPIP are, e.g., providing load balancing in the cloud domain or shortest delay for the services. Load balancing is realized by selecting the DC with the lowest current load as the DR site without considering the network latency. In the latter strategy, the load is not considered and the DC providing the shortest latency to the primary site is selected.

As shown in Figure 6.1b, the virtual network is connected to only one DC over the virtual path chosen by the VNO. Providing resilience is the responsibility of the dcPIP and it is realized by reserving redundant resources in the PIP domain and redirecting the traffic to the DR site in the physical network in case of failure, i.e., in case of failure of VM1, the traffic will be routed on $l_p$ and $l_c$ to VM2. Note that in this case the path $l_c$, node 4 and VM2 are transparent to the VNO perspective. However, on the PIP side, the mapping of the virtual link $l_p$ has to be extended to reach the DR site, i.e., the mapping of $l_p$ is changed in the internal database of the PIP as the physical paths of $l_p + l_c$ for the failure case. Hence, in case of failure at VM1, the recovery actions are solely taken by the PIP and the virtual topology remains unchanged from the VNO point of view, i.e., the recovery is ideally transparent to the VNO.

Finally, it should be noted that for PIP-Resilience, the number of virtual links and nodes, which have to be established and maintained in the virtual layer is lower than the VNO-Resilience as seen comparing the Figures 6.1a and 6.1b.

#### 6.2.1.3 Simulation Framework

The simulations are performed using a virtual network simulation tool described in Chapter 4. This tool enables the generation of random physical and virtual network topologies in form of graphs. The aim of the simulations is to compare the maximum propagation delay,

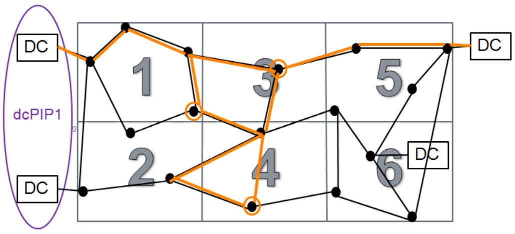


Figure 6.2: The physical network is divided to equal size availability regions, where the failure in one regions is assumed not to affect the other regions. The DCs belong to the same region as their network connection nodes.

which can be guaranteed in the two different resilience design scenarios, namely for VNO-Resilience and PIP-Resilience in case of DC failures. The simulation results are obtained with a confidence level of 95% with ±5% confidence interval.

The simulations are performed for one (merged) nPIP network with random dcPIPs connected to it. The simulation has three nested loops. The first one is for the virtual networks, i.e., for a random nPIP network and a random set of dcPIPs, random virtual networks are generated. For each virtual network, we assume to have a DC service request from each virtual node. To satisfy these requests primary and DR DCs sites are chosen for both scenarios using (6.1) and (6.2) for the average and maximum latency calculation, respectively, and the services are routed in each scenario to the corresponding DCs. The worst-case propagation delay occurring in the virtual network for each scenario is calculated and compared. A new random virtual network is generated and its cloud connection request is processed until the required confidence level on the delay difference is reached. Then, a new set of dcPIPs is generated and the virtual network simulation is repeated for this set. The second loop stops after the required confidence level on delay difference is reached for different random dcPIP sets. Similarly, this second step is repeated for each random nPIP network until the required confidence level on the whole simulation result is reached.

The nPIP networks are generated based on the algorithm described in [87]. The physical network is then divided into availability regions as shown in Figure 6.2 assuming that a failure in one availability region would not affect the other regions. Depending on the failure cause, against which the network needs to be protected, an appropriate availability region size is chosen. According to [136], a ”far enough” distance would be 105 miles, which can offer geographical disjointness even in case of hurricanes. We use 6 availability regions for a 30 nodes physical network, 12 for 60 nodes and 20 for 100 nodes respectively. The DCs of each dcPIP are placed so that each of them is in a different availability region. The placement of the DCs can occur randomly or by choosing the locations to be as far away as possible from each other, which enables the cloud providers to access and serve a larger part of the physical network with lower latency. For the simulations, the number of dcPIPs, the number of DCs per dcPIP and their placement strategy are specified as input parameters as listed in Section A.1.2. This allows us to compare the two models for various DC placements and ownership options. Moreover, similar to the unicast services, a uniform demand matrix is used, where this time a unit service is requested from each service source node, which might be routed to any available DC.

In the simulations two dcPIP strategies are implemented. In case of Shortest Delay, the dcPIP chooses the closest DC to the primary DC as the DR site. For load balancing, the DC with the lowest current load is chosen as the DR site. However, for the simulations the load of the DCs is assumed to be the same, which causes the DCs to be chosen randomly. Hence, it is named the Random Selection strategy.

For virtual network generation, the number of the virtual nodes is given as a simulation parameter. For the simulations, it is assumed that the virtual network already exists and new cloud connections are requested. Upon the arrival of these requests, depending on the scenario, either the primary and DR DC sites or only the primary DC is chosen and the services are routed. The DCs can be chosen to minimize either the average delay for all the running cloud services or the maximum delay of the virtual network.

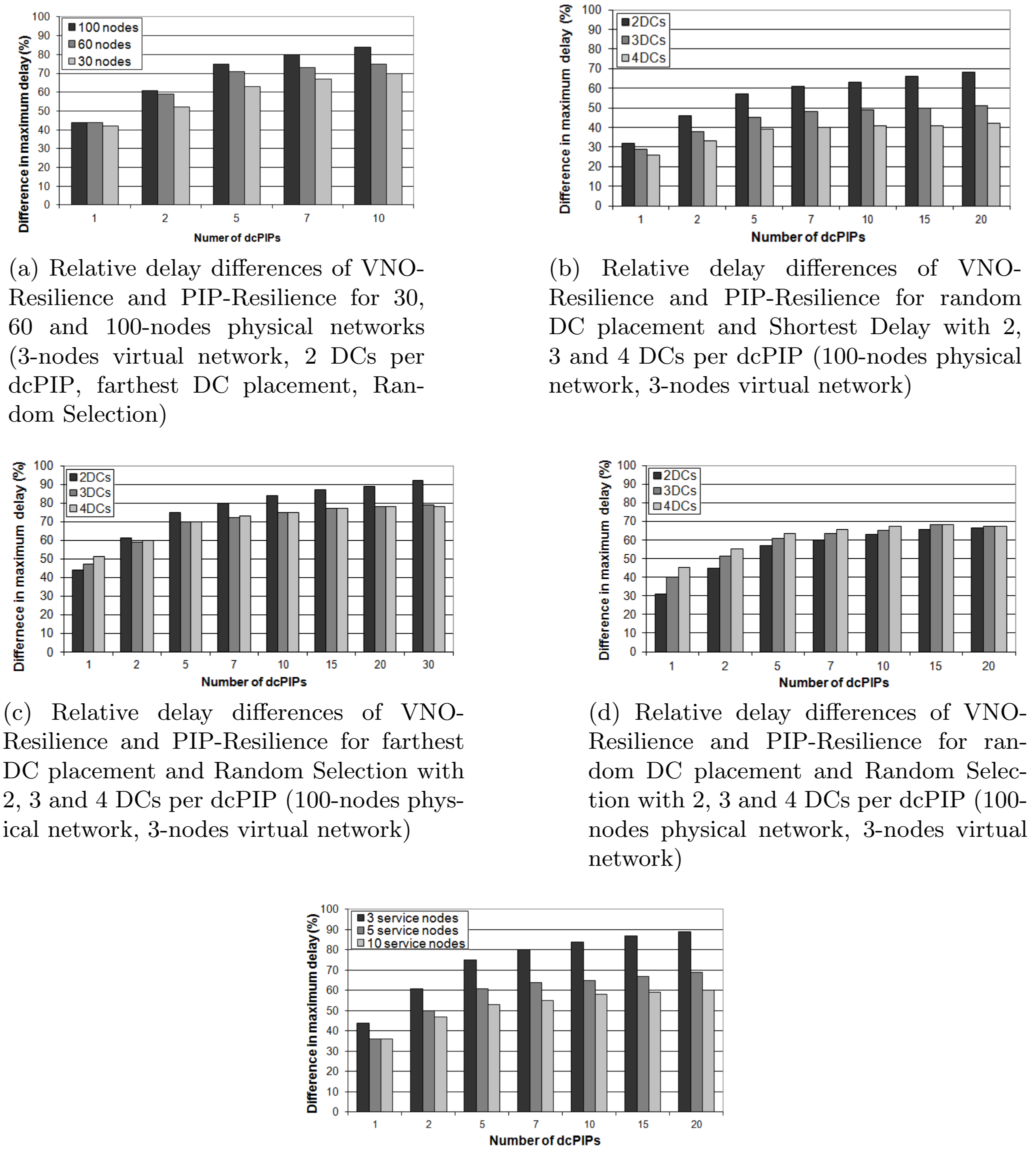
#### 6.2.1.4 Simulation Results and Evaluation

In this section, we present the most important outcomes of the simulations. Using average and maximum propagation delay as the DC selection metric provides similar results. Guaranteeing a certain maximum delay value can be of higher interest to a PIP or VNO in defining the QoS they provide or for the Service Level Agreements (SLAs). Therefore, we will focus on the results for the maximum delay in this section.

In the simulations, the physical networks are generated in a square area with $e_G = 100$ unit length edges. Average nodal degree is in the interval $[2, 4]$ and maximum nodal degree per node is 5. For national size networks, one unit length corresponds to 10 km. In this case, for two DCs with farthest location, 100-nodes physical networks and 10-nodes virtual networks, the simulation result for maximum round-trip propagation delay is 35 ms. Note that, increasing the area of the network where $e'_G = x * e_G$, would increase the delay as well $x$ times. Thus, e.g., for a European size network the maximum propagation delay would be already $> 100$ ms. For the rest of the paper, the relative delay difference of PIP-Resilience and VNO-Resilience with different parameters is analyzed. The results are calculated by taking the difference of PIP-Resilience and VNO-Resilience delay values and dividing it by the VNO-Resilience delay value.

For a given number of dcPIPs, number of DCs per dcPIP and the same PIP-Resilience strategy, the same difference in maximum delay is observed for all tested physical networks if the DC locations are chosen randomly. However, with the "farthest" option, a larger network provides higher delay difference for the two scenarios and this effect increases slightly with the number of the dcPIPs as shown in Figure 6.3a. For 10 dcPIPs, PIP-Resilience results in 85% higher delay compared to VNO-Resilience in 100-nodes physical networks and this is reduced to 70% if the physical network has only 30 nodes. The reason for that is in a larger network, with a larger number of availability regions, the distances between the DCs are longer compared to a smaller network, which causes a higher delay difference. Moreover, an increase in the number of the dcPIPs offers more options to a VNO for DC selection and, as a result, increases the delay difference between the two scenarios. However, this effect saturates with a sufficiently high number of dcPIPs and this point is reached by a smaller network earlier. Comparing Figures 6.3a and 6.3c it is seen that this point lies for the 30-nodes physical networks around 5 dcPIPs, for 60-nodes around 7 and for 100-nodes around 20 dcPIPs.

The case with one DC per dcPIP is excluded from the simulations since in that case PIP-Resilience is not possible at all. This already shows the benefit of intelligent routing making use of the overall view a VNO can have on the combined resources of nPIPs and dcPIPs. In case of a failure of the primary DC, with VNO-Resilience the services are routed over the link $l_b$ to the DR site as shown in Figure 6.1a. However, in PIP-Resilience such an optimization is not possible and the services have to be redirected from node 3 to the DR site using the link $l_c$ as shown in Figure 6.1b. For two far located DCs, this causes around 40% difference in the guaranteed maximum delay for the two scenarios. Moreover, starting with two dcPIPs, the delay difference goes over 50% for all the test networks. As shown in Figure 6.3c, the delay difference of the two scenarios exceeds 90% for a high number of available dcPIPs.



(a) Relative delay differences of VNO-Resilience and PIP-Resilience for 30, 60 and 100-nodes physical networks (3-nodes virtual network, 2 DCs per dcPIP, farthest DC placement, Random Selection)

(b) Relative delay differences of VNO-Resilience and PIP-Resilience for random DC placement and Shortest Delay with 2, 3 and 4 DCs per dcPIP (100-nodes physical network, 3-nodes virtual network)

(c) Relative delay differences of VNO-Resilience and PIP-Resilience for farthest DC placement and Random Selection with 2, 3 and 4 DCs per dcPIP (100-nodes physical network, 3-nodes virtual network)

(d) Relative delay differences of VNO-Resilience and PIP-Resilience for random DC placement and Random Selection with 2, 3 and 4 DCs per dcPIP (100-nodes physical network, 3-nodes virtual network)

(e) Relative delay differences of VNO-Resilience and PIP-Resilience for random DC placement and Random Selection with virtual networks having 3, 5 and 10 nodes (100-nodes physical network, 2 DCs per dcPIP)

Figure 6.3: Maximum delay performance comparison of VNO-Resilience and PIP-Resilience for cloud connections under different circumstances

As shown in Figure 6.3, the delay difference increases with increasing number of dcPIPs. However, the effect of the increasing number of DCs per dcPIP depends on the PIP-Resilience strategy. As shown in Figure 6.3b, with Shortest Delay strategy the delay difference decreases drastically with increasing number of DCs for the same dcPIP amount. The reason for this is that when the DC pool of a dcPIP enlarges, the probability that the two chosen DCs will be closer to each other increases as well, which shortens the backup path for the PIP-Resilience and, thus decreases the difference. However, using Random Selection, since the DR site is chosen randomly for PIP-Resilience, this effect gets smaller. For a single dcPIP a slight opposite effect is observed as shown in Figure 6.3d, since the maximum delay in PIP-Resilience slightly increases with increasing number of DCs.

Comparing Figures 6.3c and 6.3d, it is seen that an increase of more than 20% can be observed in the delay difference for the two scenarios when the DCs are located far away from each other instead of using the random placement. A similar effect is observed for all the physical test networks.

The effect of the virtual network topology on the difference of the maximum delay in the two scenarios is shown in Figure 6.3e. It is observed that the relative delay difference of the two resilience designs decreases with an increasing number of service source nodes in a virtual network, where 30% decrease is observed if 10-nodes virtual networks are used instead of 3-nodes virtual networks with 20 dcPIPs. Meanwhile, the absolute delay difference slightly increases as a result of higher absolute delay values. The maximum delay is caused by the service, which has the farthest source node to the VM location. When the number of the service nodes (virtual nodes) increases, the service with the highest latency has to traverse a longer distance already inside the virtual network, which in turn decreases the observed delay difference of the two scenarios.

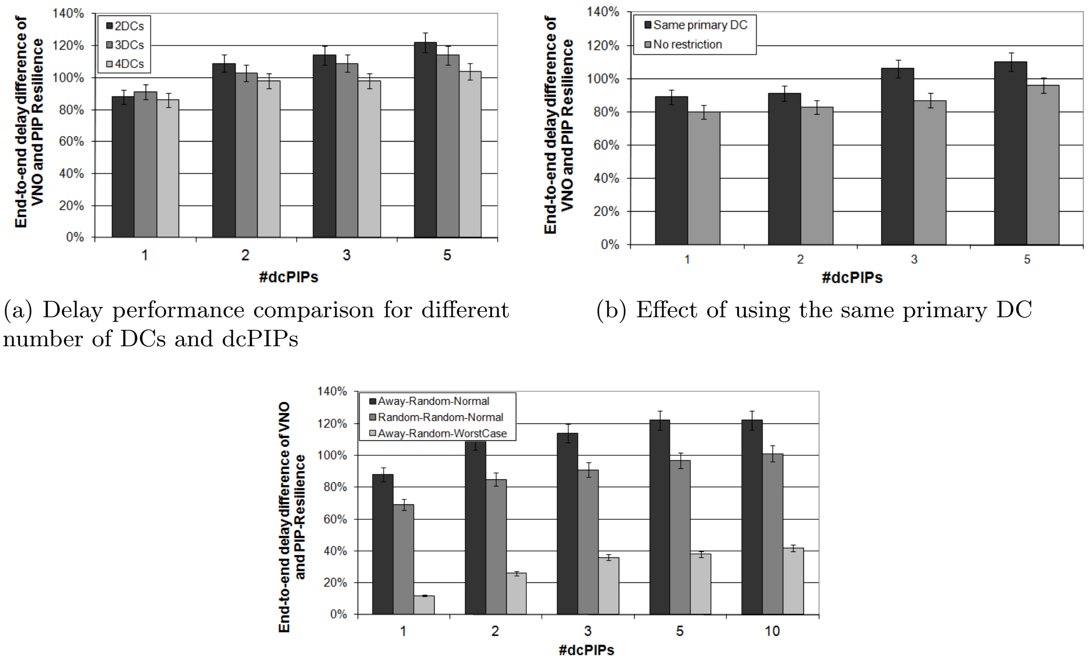
We also perform simulations, where the average propagation delay is used as the DC selection criterion. They provide results having the same trends and result in values in the same range as the presented maximum delay simulations. For 100-nodes physical network, 3-nodes virtual network, 20 dcPIPs with farthest DC location and Random Selection strategy, the delay difference of the two designs is 86%. This result shows that the effect of the optimization strategy used by DC selection on the relative maximum latency performance ratio of the two resilience designs is negligible.

### 6.2.2 Extending Resilient Virtual Networks for Cloud Services

In this subsection, DC connection models for resilient virtual networks are introduced and their performance is evaluated. The virtual network design is performed according to Chapter 5. Therefore, the main difference with the preceding subsection is that the virtual network topologies in VNO-Resilience and PIP-Resilience cases are different since they are designed with having resilience in the virtual layer and in the physical layer, respectively. Similar to the previous subsection, the existing virtual network is connected to primary and DR sites to serve all the cloud services within the virtual network. The design aim of the models is providing resilience in presence of both network and DC failures. The cloud connection models for VNO-Resilience and PIP-Resilience cases with the given corresponding virtual network topologies follows the design described in the previous subsection.

#### 6.2.2.1 Delay Performance Evaluation of the End-To-End System

The end-to-end maximum delay performance is evaluated by combining corresponding virtual network designs from Chapter 5 with the resilient DC connection models. We compare the delay performance of the models under different conditions like varying virtual network size, number of available DCs, number of different dcPIP domains, location of the



(a) Delay performance comparison for different number of DCs and dcPIPs

(b) Effect of using the same primary DC

(c) Effect of random and worst-case options

Figure 6.4: Performance comparisons of DC connection models

DCs, different DC connection model preferences and different failure cases. Similar to the previous subsection, the DCs can be placed either randomly, with the option ”random”, or to obtain maximum distance between them, namely with the option ”farthest.” Note that for both cases, the DCs of a dcPIP are located in different availability regions. Finally, it is assumed that in PIP-Resilience, the dcPIP chooses the DR site randomly from its domain due to the assumption of equally loaded DCs. The list of all the used parameter settings is provided in Section A.1.3. Differently than the previous subsection, only two DCs are used to observe the effect of this basic case and to focus on the difference of the models.

In the simulations virtual networks are generated for randomly selected service source nodes on physical topologies with DCs located randomly on them. Note that the chosen test networks, NobelUS and NobelEU [100], are realistic topologies covering large physical areas. This enables end-to-end resilience design even in case of disasters and makes the problem more interesting by possibly enabling having multiple PIPs. For each virtual network and DC set, the cloud connections are designed using the two models and the maximum end-to-end latency observed in both cases is compared until the confidence level of 95% with a ±5% confidence interval is reached for the result.

Figure 6.4a shows the effect of the number of the different dcPIP domains and amount of DCs each domain possesses on the end-to-end maximum delay difference of PIP-Resilience and VNO-Resilience. The results are obtained using 3-nodes virtual networks mapped on the NobelEU network with random DCs. For this simulation the DC location option is ”farthest” and the same primary DC is used for PIP-Resilience and VNO-Resilience. It is observed that PIP-Resilience results always in higher end-to-end delay compared to VNO-Resilience and this difference increases with increasing number of dcPIPs. However, for a certain number of dcPIPs, increasing the amount of DCs per dcPIP decreases the relative delay difference, since the dcPIPs' DC selection options increase as well.

The simulations performed with the NobelUS network show that if in PIP-Resilience the primary DC is selected freely to minimize the latency, the relative delay difference is decreased by 10% compared with the same primary DC selection scenario as shown in Figure 6.4b. Moreover, comparing Figures 6.4a and 6.4b, it is seen that a larger physical network results in higher relative delay difference. For NobelUS network, with a single dcPIP and two DCs, the absolute maximum end-to-end round-trip delay of the PIP-Resilience is around 112 ms for 3-nodes virtual networks. For NobelEU network, the maximum round-trip delay of 3-nodes virtual networks is 107 ms and of 5-nodes virtual networks 117 ms. However, the relative delay difference of the PIP-Resilience and VNO-Resilience remains almost constant for different virtual network sizes.

Finally, different DC location and protection options are compared using the NobelEU network and 3-nodes virtual networks as shown in Figure 6.4c. In all cases PIP-Resilience results in a higher maximum delay compared to VNO-Resilience. This difference goes beyond 120% if more than 5 dcPIPs are available for the "farthest" DC location option. When the DCs are placed randomly, the relative delay difference is reduced by around 20%. Finally, in a worst-case scenario, the relative delay difference is drastically decreased and reaches 40% for 10 dcPIPs.

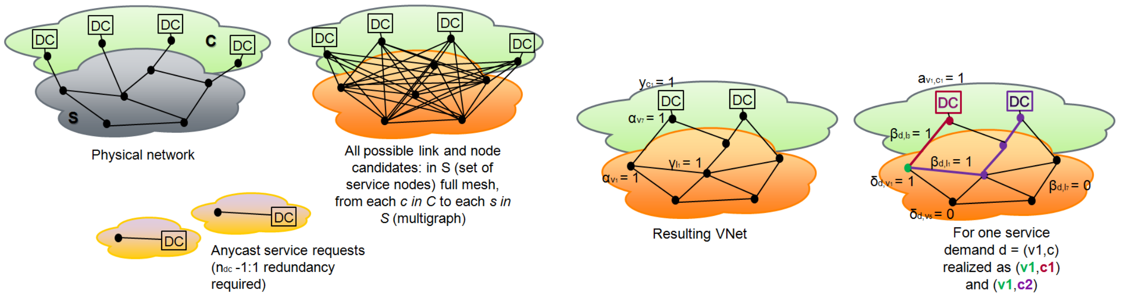
### 6.2.3 Summary of Resilient Cloud Connection Models for Virtual Networks

This section aimed providing a first analysis for the selection of the resilience layer for cloud connections, namely the virtual or physical layer resilience. Two scenarios are evaluated where in the first one the same virtual network is used for both models and in the second one resilient virtual networks are generated according to Chapter 5 for the corresponding resilience options. Both scenarios show a clear benefit of using virtual layer resilience for cloud connections. The benefit of the virtual layer resilience is twofold. Routing directly to two DC sites in the virtual layer reduces the service latency compared with the physical layer resilience, where the services are redirected to the DR site from the primary site in case of a failure in the primary site. Secondly, the dcPIPs are limited to their own cloud domains when selecting a DR site, whereas a VNO can select the primary and DR sites among the available DCs of different cloud providers. Our quantitative study shows that the latency gain of virtual layer resilience compared with having it in the physical layer is about 60% if the same virtual network is used for both models and reaches 120% for virtual network topologies designed with the corresponding resilience options for the same settings and similar size physical network topologies used in the simulations. The main reason behind the differences of the results of these two cases is the topological dissimilarities like number of virtual links, routing of the services etc. caused by optimization using the VNO-Resilience and PIP-Resilience models in the latter.

## 6.3 Combined Optimization for Virtual Network Design

This section introduces mathematical models for resilient virtual network design with different resilience options, namely VNO-Resilience and PIP-Resilience, having resilience in the virtual or physical layer, respectively, for both network and DC resources. The virtual network topology is not pre-given and is directly optimized for cloud service requests. Firstly, the general MILP model without resilience is described and then the details of each resilience option are given.

The MILP takes as **input** the (i) undirected physical network graph $G = (N, E)$, (ii) available DCs with their connection nodes $c \in C \subseteq N$, (iii) set of virtual link and node candidates given as a multi-graph $G_l = (V, L)$, where we have a 1-to-1 mapping for the virtual nodes but parallel virtual links with different physical path mappings can exist between a node pair to maintain linearity, and (iv) a set of cloud (anycast) service requests as shown in Figure 6.5a. The **objective** is to find a resilient virtual network topology


DC
DC
DC
C
DC
S
Physical network
DC
DC
DC
DC
All possible link and node candidates: in S (set of service nodes) full mesh, from each c in C to each s in S (multigraph)
DC
DC
Anycast service requests ($n_{dc}$ -1:1 redundancy required)
$y_{c_1} = 1$
DC
DC
$\alpha_{v_7} = 1$
$\gamma_{l_1} = 1$
$\alpha_{v_1} = 1$
Resulting VNet
$a_{v1,c_1} = 1$
DC
DC
$\beta_{d,l_3} = 1$
$\beta_{d,l_1} = 1$
$\delta_{d,v_1} = 1$
$\beta_{d,l_7} = 0$
$\delta_{d,v_5} = 0$
For one service demand d = (v1,c) realized as (v1,c1) and (v1,c2)

(a) Inputs to the MILP: The physical network topology and DC sites, all virtual link and node candidates and the cloud service requests

(b) Outputs of the MILP: Virtual network topology with its mapping and service routing

Figure 6.5: The input and output of the proposed optimization models

with attached DCs either (i) with a minimum virtual network setup cost, which is the fee a VNO needs to pay to the PIP(s) for the rental of the selected virtual resources and the establishment of the virtual network, or (ii) with a minimum average service delay within the virtual network **such that** all cloud service requests are satisfied. The output of the MILP is presented in Figure 6.5b.

In both resilience options, each anycast demand (cloud service) from a service source node to the cloud is routed to $n_{dc}$ servers, where a single one is operational at a time ($n_{dc}$-1:1 redundancy). The primary and DR sites are chosen per service from the set of all available and suitable DCs. The DC sites are modeled by their network connection points. Our resilient virtual network design models offer resilience in presence of both DC and network failures.

### 6.3.1 General Model without Resilience

Firstly, the general virtual network design model for cloud services is introduced, which is the basis for both of the proposed models. Then, for each model, the differences to the basic model are presented. In the following, a list of the sets, parameters and variables used in all models is presented.

- *Sets:*
  - $S$: Set of the service nodes
  - $C$: Set of the DC connection nodes
  - $V$: Set of all the virtual nodes with $S \cup C = V$ and $S \cap C = \{\}$
  - $L$: Set of the virtual link candidates, where there is at least one link between all node pairs in $S$ and from each node in $S$ to all nodes in $C$
  - $D_u$: Set of all the possible unicast realizations of the requested cloud services, where $|D_u| = |S| \cdot |C|$ and $d = (s, c) \in D_u$ with $s \in S$ and $c \in C$
  - $D_s$: Set of all the possible unicast realizations of the requested cloud service having the source node $s \in S$ with $|D_s| = |C|$ and $D_s \subseteq D_u$
  - $E_l$: Set of the endpoints of a virtual link $l \in L$
  - $Z$: Set of virtual link pairs $(l, k) \in L^2$, which are not disjoint
  - $E$: Set of the edges in the physical network topology
  - $N$: Set of the nodes in the physical network topology
  - $P_l$: Set of the physical edges $e \in E$, on which the virtual link $l \in L$ is mapped
  - $R$: Set of DC connection node pairs $(c_1, c_2) \in C^2$ with $c_1 \neq c_2$, which are located in the same availability region of the physical topology
- *Parameters:*
  - $n_{dc}$: Number of the DCs, which will be selected for each cloud service with $n_{dc} \in \{1, ..., |C|\}$

  - $b_d$: Requested bandwidth for the service $d \in D_u$
  - $n_d$: Requested network node resources for the service $d \in D_u$
  - $r_d$: Requested server resources for the service $d \in D_u$
  - $t_l$: Physical length of the virtual link $l \in L$
  - $\lambda_l$: Fixed setup cost for having a new virtual link $l \in L$ in the virtual network
  - $\theta_l$: Setup cost per unit capacity of a virtual link $l \in L$
  - $\mu_v$: Fixed setup cost for having a virtual network node $v \in V$ in the virtual network
  - $\eta_v$: Setup cost per unit capacity of a virtual network node $v \in V$
  - $\phi_c$: Fixed setup cost for having a new virtual machine in the virtual network, which is connected to node $c \in C$
  - $\varphi_c$: Setup cost per unit capacity of a virtual machine connected to node $c \in C$
- *Variables:*
  - $a_{s,c}$: Binary variable taking the value of 1 if a virtual machine is placed into the DC connected to node $c \in C$ to satisfy the anycast demand with source $s \in S$, 0 otherwise
  - $\beta_{d,l}$: Binary variable taking the value of 1 if the link $l \in L$ is used for the demand $d \in D_u$ and if demand $d = (s, c)$ is chosen as one of the realizations of the cloud service with source $s \in S$, 0 otherwise
  - $\delta_{d,v}$: Binary variable taking the value of 1 if the node $v \in V$ is used for the demand $d \in D_u$ and if demand $d = (s, c)$ is chosen as one of the realizations of the cloud service with source $s \in S$, 0 otherwise
  - $\gamma_l$: Binary variable taking the value of 1 if the link $l \in L$ is included to the virtual network, 0 otherwise
  - $\alpha_v$: Binary variable taking the value of 1 if the node $v \in V$ is included to the virtual network, 0 otherwise
  - $y_c$: Binary variable taking the value of 1 if a virtual machine on the DC connected to node $c \in C$ is included to the virtual network, 0 otherwise
  - $u_l$: Used capacity on link $l \in L$ with $u_l \in [0, \infty)$
  - $\omega_v$: Used capacity on node $v \in V$ with $\omega_v \in [0, \infty)$
  - $z_c$: Used capacity on DC connected to node $c \in C$ with $z_c \in [0, \infty)$

As mentioned before, there are two objective functions defined for different optimization objectives, namely virtual network cost minimization and propagation delay minimization. The cost of the virtual network constitutes of link cost, network node cost and VM cost as given in (6.3), (6.4) and (6.5), respectively.

$$\varepsilon_l = \lambda_l \gamma_l + \theta_l u_l \quad \forall l \in L \tag{6.3}$$

$$\varepsilon_v = \mu_v \alpha_v + \eta_v \omega_v \quad \forall v \in V \tag{6.4}$$

$$\varepsilon_c = \phi_c y_c + \varphi_c z_c \quad \forall c \in C \tag{6.5}$$

Each of these costs has two parts, as already described in Section 5.5, namely the fixed setup cost for having a new link, node or VM in the virtual network and the capacity dependent cost depending on the requested capacity of a link, node or VM. For sufficiently simple PIP-VNO business relationships, a linear cost model is assumed. For cost minimization, the overall setup cost of the virtual network is minimized as given in (6.6).

$$\min\ \varepsilon,\ \varepsilon = \sum_{l \in L} \varepsilon_l + \sum_{v \in V} \varepsilon_v + \sum_{c \in C} \varepsilon_c \tag{6.6}$$

For propagation delay minimization, the total length of the routes for each service is minimized. Assuming that the network is designed for normal load conditions, we only consider the propagation delay of the physical routes as the latency metric for a service. The delay minimization objective function is given in (6.7).

$$\min \quad \sum_{d \in D_u} \sum_{l \in L} \beta_{d,l}\, t_l \tag{6.7}$$

The main constraints of the virtual network design model for cloud services are given in the following. Equation (6.8) ensures that $n_{dc} \in \{1, ..., |C|\}$ server locations are chosen for a cloud service with source $s$. $n_{dc} = 1$ means that there is no DC resilience, i.e. no protection against DC failures, in the virtual layer. Increasing $n_{dc}$ increases the level of protection.

$$\sum_{c \in C} a_{s,c} = n_{dc} \quad \forall s \in S \tag{6.8}$$

Equation (6.9) is the unsplittable flow conservation constraint ensuring that all the flows entering a node also leave that node if it is an intermediate node and there is only one flow entering or leaving the node for one service if that node is the target or the source of that service, respectively. Equation (6.10) ensures that a node is flagged as ”used” for a service if it is the source or the target of that service and if it is chosen as a realization of the cloud service with source $s$.

$$\sum_{l \in L: v \in E_l} \beta_{d,l} = \begin{cases} a_{s,c} & \text{if } v = s \text{ or } v = c \\ 2\delta_{d,v} & \text{otherwise} \end{cases} \qquad \forall d = (s,c) \in D_u,\ v \in V \tag{6.9}$$

$$\delta_{d,v} = a_{s,c} \quad \forall d = (s,c) \in D_u,\ v \in \{s, c\} \tag{6.10}$$

Constraints (6.11), (6.12) and (6.13) state that if a virtual link, node or VM carries the traffic of at least one service, it is part of the resulting virtual network, and otherwise not.

$$\gamma_l \geq \beta_{d,l} \quad \forall l \in L,\ d \in D_u \tag{6.11}$$

$$\alpha_v \geq \delta_{d,v} \quad \forall v \in V,\ d \in D_u \tag{6.12}$$

$$y_c \geq a_{s,c} \quad \forall c \in C,\ s \in S \tag{6.13}$$

Additionally, (6.14), (6.15) and (6.16) provide upper bounds for $\gamma_l$, $\alpha_v$ and $y_c$, respectively, which ensures that a virtual link, node or VM is part of the resulting virtual network only if it is actually used by some service. These bounds are only necessary for calculating the virtual network cost in delay optimization to obtain meaningful cost values but do not restrict the optimality.

$$\gamma_l \leq \sum_{d \in D_u} \beta_{d,l} \quad \forall l \in L \tag{6.14}$$

$$\alpha_v \leq \sum_{d \in D_u} \delta_{d,v} \quad \forall v \in V \tag{6.15}$$

$$y_c \leq \sum_{s \in S} a_{s,c} \quad \forall c \in C \tag{6.16}$$

Finally, (6.17), (6.18) and (6.19) are the constraints for calculating the required virtual link, node and VM capacities, respectively. The required capacities are calculated by summing up the values of the demand requests utilizing the corresponding virtual link, node or VM.

$$u_l \geq \sum_{d \in D_u} \beta_{d,l}\, b_d \quad \forall l \in L \tag{6.17}$$

$$\omega_v \geq \sum_{d \in D_u} \delta_{d,v}\, n_d \quad \forall v \in V \tag{6.18}$$

$$z_c \geq \sum_{s \in S} a_{s,c}\, r_d \quad \forall c \in C \text{ with } d = (s,c) \tag{6.19}$$

Unicast service requests can be easily included into the model by extending the service set and by adding the unicast flow constraint as shown in Chapter 5. However, we have omitted the inclusion of the unicast services in this section since we focus on the combined optimization of network and IT resources.

### 6.3.2 VNO-Resilience

In VNO-Resilience, the virtual network is designed for a given set of services, which are routed in the virtual layer to $n_{dc}$ different DC site locations. We assume $n_{dc} = 2$ as a practical number in this thesis for the simulations. For this model, both the DC sites and the paths leading to the DC sites have to be physically disjoint, such that in case of a failure at the primary site, the DR site can take over by re-routing the service inside the virtual network. Therefore, diversity constraints are needed to be added for these paths and DC sites to the model.

Constraints (6.20) and (6.21) ensure link and node-diversity respectively for the connection paths. Additionally, in case of node-diversity, node-disjointness in the physical layer has to be ensured by extending the set $Z$ accordingly.

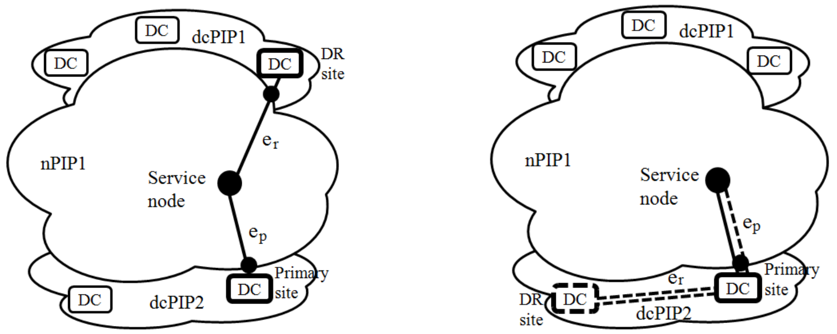
$$\beta_{d_1,l} + \beta_{d_2,k} \leq 1 \quad \forall s \in S,\ (d_1,d_2) \in D_s^2,\ (l,k) \in Z \tag{6.20}$$

$$\begin{aligned} \delta_{d_1,v_1} + \delta_{d_2,v_2} \leq 1 \quad \forall s \in S,\ &(d_1,d_2) \in D_s^2, \\ &(v_1,v_2) \in (V \setminus \{s\})^2 \end{aligned} \tag{6.21}$$

Furthermore, we need to make sure that the primary and DR sites are located in different availability regions as given in (6.22). The diversity constraints can be easily extended for multiple and regional failures by generating the set $R$ accordingly.

$$a_{s,c_1} + a_{s,c_2} \leq 1 \quad \forall s \in S,\ (c_1,c_2) \in R \tag{6.22}$$

Figure 6.6a shows the realization of VNO-Resilience for a single service node. For both primary and DR sites, the connection nodes of the corresponding DCs as well as the paths connecting them to the source node of the cloud service, $e_p$ and $e_r$, are part of the virtual network. The paths $e_p$ and $e_r$ can be composed of multiple virtual links and nodes and to ensure resilience they have to be physically disjoint.



(a) VNO-Resilience: Each service node is connected in the virtual layer to two DC sites.

(b) PIP-Resilience: Each service node is connected to a single DC site in the virtual layer. Resilience is provided by the PIPs.

Figure 6.6: Proposed resilience models at the virtual and physical layers

### 6.3.3 PIP-Resilience

In case of PIP-Resilience, providing resilience is the responsibility of the nPIPs and dcPIPs. The services are routed on a single path in the virtual layer to the primary server site, i.e. $n_{dc}$ is taken equal to 1 in (6.8). This virtual path is protected by the corresponding nPIP(s), where each virtual link has a 1:1 protection mapping on the physical layer. Since 1:1 protected path mapping is given as an input to the MILP, diversity constraints are unnecessary. Furthermore, the dcPIP owning the primary site is responsible for providing DC resilience. The realization of PIP-Resilience for a single service is shown in Figure 6.6b. Similar to the VNO-Resilience case, a single DR site is assumed, which can be easily generalized for $n_{dc}$ DR sites. From the VNO perspective, only the connection path to the primary site, the network connection node of the primary site and the primary site VM are visible inside the virtual network. Upon failure of the primary site, the services are re-routed to the DR site in the physical layer, where the virtual network and the routing of the services in the virtual network remain unchanged.

In PIP-Resilience, the DR site(s) for each primary DC candidate and their resilient physical connection path(s) are pre-calculated. This information is incorporated in the fixed cost factor of the corresponding primary VM. Different strategies can be implemented to select a DR server site as described in the previous sections. It can be e.g. chosen randomly or such that it offers the shortest interconnection path. Once the DR site is chosen, their interconnection path is calculated as the shortest disjoint path pair connecting the two sites. Finally, the physical length $t_l$ of a virtual link $l \in L$ is set as the total length of the physical working and protection paths of the virtual link $l$.

## 6.4 Performance Evaluation of the Proposed Models

In this section, the performance evaluation of the proposed models is presented. The section is divided into three subsections, where first the used simulation framework and parameters are introduced briefly. Afterwards, the simulation results are presented. Finally, a discussion about the implementation and applicability of the models is provided.

### 6.4.1 Simulation Framework and Parameters

We performed simulations to provide insights into the applicability and efficiency of the proposed models compared with prior approaches and finally to present a quantitative analysis for the effect of different parameters and cost factors on the performance of the models. In this subsection the simulation framework and the used parameter settings are shortly presented. The simulations are performed using the Java Virtual Network Simulator, which is described in Chapter 4. The optimization problems are implemented using the IBM Concert library and solved with CPLEX 12.3. The resulting optimal virtual networks for different settings are then simulated to determine the maximum propagation delay, which can be guaranteed for that virtual network, and the virtual network cost. The simulation results are within a ±5% confidence interval at a confidence level of 95%.

We compare the performance of our models with two models where resilient routing is provided in the virtual network and the virtual link mapping is fixed as the shortest path mapping. The first model, namely the Shortest Path Mapping (SPM) model, uses the set of service nodes and DC connection nodes as the initial virtual node set like in VNO-Resilience and PIP-Resilience. For the second model, namely the SPM with Additional Nodes (SPMwAN) model, we use an extended initial virtual node set, in which a virtual node corresponding to each physical node is included. Thus, the optimal virtual network can include some virtual nodes, which are not used for services but just for routing purposes.

The simulations are performed using the physical network topologies NobelUS and NobelEU [100]. NobelUS has 14 nodes and 21 edges. NobelEU has 28 nodes and 41 edges. At each simulation run, for the given physical network topology, the DCs are placed in the network. The simulator takes as input the number of dcPIPs, number of DCs per PIP and the DC placement strategy, which places the DCs randomly or as far as possible from each other. For both cases, we divide the physical topology map into equal-sized rectangular availability regions. A failure in one region is assumed not to affect the other regions when the size of the regions is adjusted accordingly [136]. Availability regions enable DC resilience against natural disasters like hurricanes, tsunamis, earthquakes, floods etc., where DCs of a single dcPIP are placed such that each DC is in a different availability region. We used 12 regions for the NobelEU and 6 for the NobelUS topology. A complete list of simulator parameters is provided in Section A.1.4. In these simulations, we vary both the number of the service nodes, as well as the number of DCs and dcPIPs to observe the effect of different load and DC placement situations. As discussed in the former sections, we use a uniform demand matrix, where there is a service request from each service node to the cloud.

For the given physical network and selected DC locations, we generate random cloud service requests, where the number of services is given as an input parameter, and the service nodes are chosen randomly from the physical topology. Afterwards, an optimal resilient virtual network is calculated according to the selected resilience method. Note that for the NobelUS topology, the worst-case duration of solving the MILP is around 1 minute for VNO-Resilience and 0.2 seconds for PIP-Resilience for the simulated cases. Depending on the simulation aim, the corresponding value, e.g. cost of the optimal virtual network or the maximum delay occurring in the virtual network is computed. We continue to generate random services and solve the models until a required confidence level is reached for the mean of the simulation aim values. This mean value corresponds to the used DC set. Afterwards, the same loop is repeated for a new random DC set, until the required confidence level is reached for the results of different DC sets.

In the remainder of this section we provide the necessary settings and formulations of virtual network setup cost with VNO-Resilience and PIP-Resilience. As given in (6.3)-(6.5) and as discussed before, the cost of a virtual network has three parts. Each of them consists of the fixed cost of placing virtual network components and the capacity-related cost depending on the size of the components. We used five cost settings, which are listed in Table 6.1.

Table 6.1: Cost factors for the virtual network design models with cloud services

| Cost setting | Link Cost | Node Cost | VM Cost |
|---|---|---|---|
| (L,1,1) | $t_l$ | 1 | 1 |
| (1,1,1) | 1 | 1 | 1 |
| (L,A,A) | $t_l$ | $a_G$ | $a_G$ |
| (1,1,A) | 1 | 1 | $a_G$ |
| (1,A,1) | 1 | $a_G$ | 1 |

$t_l$ is the resulting physical length of the virtual link in kilometers and it is used as the cost factor instead of a fixed value. In case of ”1”, the cost is constant and is one unit for the links, nodes and/or VMs. $a_G$ is the average shortest path length in a physical topology $G$ and is used as the link/node cost factors for the last three cost settings. Cost setting (L,1,1), (1,1,A) and (1,A,1) are used to evaluate the effect of the dominance of each cost factor. In setting (1,1,1), all cost factors are equal and in (L,A,A) comparable to each other. The difference of (L,A,A) compared with (1,1,1) is that the link cost depends again on the physical path length. Hence, these cost settings provide a complete list for all possible cost factor options.

In VNO-Resilience these cost factors are directly used. However, in PIP-Resilience the resilience cost needs to be included to the cost of the links and VMs. For virtual nodes no resilience is provided, and hence, the cost of the nodes remains unchanged. In PIP-Resilience, if $t_l$ is used as the cost factor for the links, resilience cost is implicitly included, since $t_l$ is the total length of the primary and backup path mappings for $l$. However, if a fixed value is used like in settings (1,1,1), (1,1,A) and (1,A,1), the additional cost of providing resilience at the physical layer should be included in the cost of a virtual link by introducing a resilience premium $r_{\text{PIP}}$ as given in (6.23).

$$\varepsilon_{l,PIP,Fixed} = (\lambda_l \gamma_l + \theta_l u_l)\, r_{\text{PIP}} \tag{6.23}$$

If DC resilience is provided by the PIP, the cost of resilience consists of the second DC site usage and the cost of the physical paths connecting the two sites. The fixed cost remains the same since neither the second VM nor the connection path is part of the virtual network. The capacity dependent cost of the resilient VM is given in (6.24) and (6.25) for the length-dependent link cost and fixed link cost cases, respectively.

$$\varphi_{c,PIP,Length} = 2\varphi_c + a_G \tag{6.24}$$

$$\varphi_{c,PIP,Fixed} = 2\varphi_c + \lambda\, r_{\text{PIP}} \tag{6.25}$$

### 6.4.2 Simulation Results

In this subsection, the simulation results are presented. First, the proposed models are compared with prior approaches and separate optimization. Then, the proposed models' performance is evaluated under different parameters and cost factors to determine their effects as well as to show under which conditions it is preferable to provision resilience at a certain layer.

#### 6.4.2.1 Comparison with Prior Approaches

We compare the performance of our VNO-Resilience model with prior approaches. Our simulations show that in around 50% of the simulation runs SPM fails to find a resilient

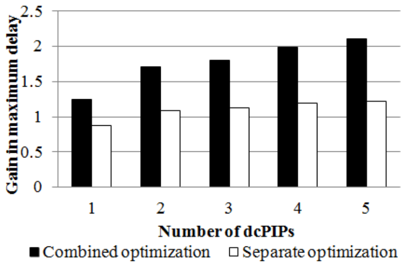


Figure 6.7: Combined vs. separate optimization over number of dcPIPs (2 DCs each, farthest DCs placement, 3 service nodes)

virtual network solution, while this value is only 0.02% for VNO-Resilience. These simulations are performed for 1 dcPIP and 2 DCs located randomly and varying the number of the service nodes between 1 and 10. For these simulations and for the delay gain simulations presented in the remainder of this section, the NobelEU topology is used, and delay optimization is applied.

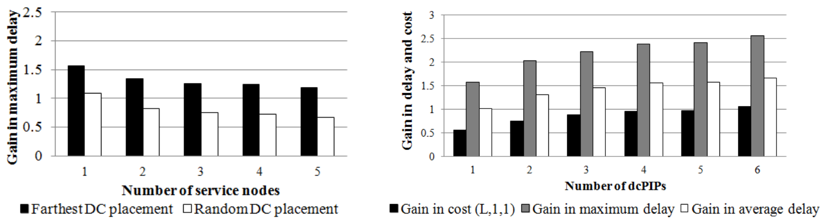
If the use of additional nodes is allowed for SPM, namely in case of SPMwAN, the simulations show that on average the resulting virtual network topology includes a higher number of virtual links and nodes compared with VNO-Resilience, where the difference is around 45% for the number of virtual links, and 40% for the number of virtual nodes. Hence, allowing additional nodes enables the prior approach to find resilient network solutions. This however increases the setup and maintenance costs of the virtual network significantly.

Figure 6.7 compares the gain of VNO-Resilience using the proposed combined optimization models with using separate optimization models introduced in Section 6.2. In the latter, the virtual network is designed for unicast services, and then it is connected to one or two DC sites by adding virtual links to it to minimize the delay to the cloud. In the former, the virtual network design optimizes network and cloud resources in combination, and it works on the service level for DC selection. Hence, combined optimization chooses a primary and a DR site per service, whereas separate optimization uses the same primary and DR sites for all services. The maximum service latency occurring in the virtual network for PIP-Resilience and VNO-Resilience, which is an important performance parameter besides the average delay performance for certain applications, is compared using the two different optimization approaches. The maximum delay is decreased for both models with combined optimization, while for VNO-Resilience this delay gain is around 50% for 5 dcPIPs and for PIP-Resilience less than 30% for the used settings. Hence, combined optimization increases the maximum delay gain compared with separate optimization as shown in Fig.6.7. With 5 dcPIPs, PIP-Resilience results in 80 ms maximum round-trip delay with combined optimization only due to the propagation and it is reduced to 25 ms for VNO-Resilience.

#### 6.4.2.2 Comparison of VNO-Resilience and PIP-Resilience

In this part, the two proposed models are compared under various conditions to determine if and how much gain one can obtain by provisioning resilience within the virtual network at the VNO layer rather than delegating it to the PIPs like in the traditional scenarios.

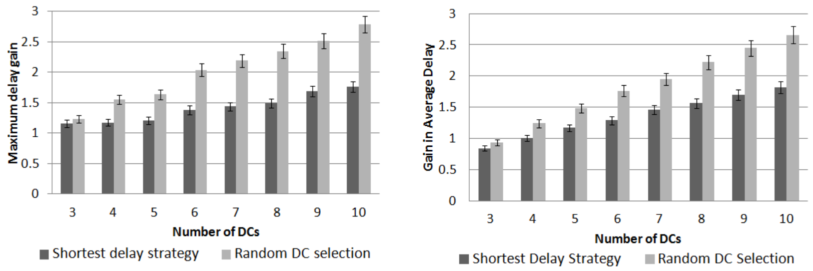
Figure 6.8a shows the effect of placing the DCs into the regions randomly vs. choosing the farthest regions on the delay gain for the maximum delay, which can be guaranteed in VNO-Resilience and PIP-Resilience. The reason for farthest DC placement would be for the dcPIP to have access to different parts of the physical topology and to increase DC resilience. Simulation results show that the farthest DC placement of the nodes increases



(a) Farthest vs. random DC placement over number of service nodes (1 dcPIP, 2 DCs each)

(b) Effect of varying the number of dcPIPs (2 DCs each, farthest DC placement, 1 service node)

Figure 6.8: Gain of VNO-Resilience over PIP-Resilience $(= \frac{x_{\text{PIP}-\text{Resilience}} - x_{\text{VNO}-\text{Resilience}}}{x_{\text{VNO}-\text{Resilience}}})$ for different models and settings



(a) Maximum delay comparison

(b) Average delay comparison

Figure 6.9: Delay gain of VNO-Resilience over PIP-Resilience $(= \frac{x_{\text{PIP}-\text{Resilience}} - x_{\text{VNO}-\text{Resilience}}}{x_{\text{VNO}-\text{Resilience}}})$ for the Shortest Delay and Random DC Selection Strategies

the maximum delay values for PIP-Resilience around 30% and for VNO-Resilience around 5-10%. Hence, the delay difference between the PIP-Resilience and VNO-Resilience is increased with farthest DC placement as shown in Figure 6.8a. Therefore, if the dcPIPs want to offer resilience for delay sensitive services, they should consider placing their DCs in a more random fashion. Moreover, it is also shown that the delay gain decreases with increasing number of service nodes. Hence, the delay gain is a more important decision parameter for smaller virtual networks.

Figure 6.8b shows the effect of the number of the dcPIPs in the network on the cost and on the maximum and average delay gain of VNO-Resilience compared with PIP-Resilience. Increasing the number of dcPIPs increases the number of the DC options for VNO-Resilience and for the primary site choice in PIP-Resilience without affecting the choice of the DR site for PIP-Resilience. Thus, the gain in average delay is expected to grow with increasing number of dcPIPs. The simulation results validate this conclusion. Moreover, it is shown that the gain in maximum delay, which can be guaranteed in the virtual network, reaches 200% already for 2 dcPIPs and increases with increasing number of dcPIPs, reaching 250% for 6 dcPIPs.

The effect of the DR site selection strategy on the maximum and average delay gain is presented in Figure 6.9a and 6.9b, respectively. Simulations performed for maximum and average delay gain show a similar trend. For both cases, the delay gain of the VNO-Resilience is reduced by almost 50% for 5 DCs and by 75% for 10 DCs using Shortest Delay Strategy instead of Random Selection. Hence, from the point of view of a dcPIPs, the choice of this strategy affects the delay performance of PIP-Resilience drastically. Therefore, shortest delay strategy should be preferred for delay-sensitive services to increase the competitiveness of the PIP-Resilience offer.

The trend of the cost and delay gain is similar for cost setting (L,1,1) as shown in Figure 6.8b. This is due to the fact that in cost setting (L,1,1), the virtual link cost is the dominant factor, which depends on the physical length of the virtual links. Hence, optimizing for cost setting (L,1,1) is aligned with delay optimization, and for this cost setting VNO-Resilience results always in cheaper virtual networks compared with PIP-Resilience.

Using cost setting (1,1,1), the cost gain depends on the choice of the resilience premium. For the simulations, the resilience premium is taken as 2. Note that for cost settings (1,1,1), (1,1,A) and (1,A,1) using cost optimization, the delay is not minimized and takes a random value depending on the selected DC site. Hence, if there are many DCs available, cost optimization for these cost settings might result in much higher delay values compared with delay optimization. If the service delay is important, delay constraints should be added to the MILP by cost optimization. This results in similar virtual network setup cost values with acceptable delay characteristics.

Cost setting (L,A,A) results in comparable cost values for PIP-Resilience and VNO-Resilience, while PIP-Resilience has slightly lower values. In this cost setting, again the physical length of the virtual links is used as the link cost factor, and hence, it also optimizes for the delay implicitly. For cost setting (1,1,A), PIP-Resilience results in lower virtual network setup cost compared with VNO-Resilience due to the higher number of VMs involved in VNO-Resilience. The cost gain of PIP-Resilience decreases with increasing number of service nodes since it causes the capacity-dependent cost of the VM to be the dominant factor compared with its initial setup cost. For one service node there is a difference of 30%, which decreases to 8% for 5 service nodes according to our simulations performed with 1 dcPIP and 2 DCs, NobelUS topology and cost optimization. For cost setting (1,A,1), where the virtual node cost is the dominant factor, PIP-Resilience always results in cheaper virtual networks since the number of virtual nodes used in VNO-Resilience and their capacity is much higher compared with PIP-Resilience. The difference in cost lies at around 40%. In Section 6.5, the hybrid resilience models are introduced, where the VNOs and PIPs share the responsibility of providing resilience. A more comprehensive comparison of all these alternatives in terms of cost, service latency, physical resource utilization and complexity is also provided in Section 6.5.3.

### 6.4.3 Implementation and Applicability

In this chapter, two novel resilient virtual network design models for cloud services are introduced, which are modeled as MILPs. We evaluated their performance in terms of virtual network setup cost and service latency under different parameters in the previous section. In this section, we will discuss the implementation possibilities and applicability of the proposed models.

The introduced models are abstracting the virtual network design from the underlying technology, and hence, they are technology independent. These models can be implemented in systems using, e.g., IP over WDM, or control planes like GMPLS or OpenFlow (OF). New interfaces are required that support the needed information exchange and control between the different roles. Regarding the interfaces, a virtual network architecture as proposed in [75] can be used. The information exchange level depends on the business model of the VNO and the PIP and on their contract. It can be expected that PIPs refrain from sharing detailed physical topological information with VNOs, while the VNOs need a certain level of information in order to be able to design virtual networks. Our proposal for an appropriate level of information exchange for both roles consists of the availability information of the virtual link candidates, the virtual nodes adjacent to the virtual link candidates, optimization related properties of the virtual link candidates and the disjointness information.

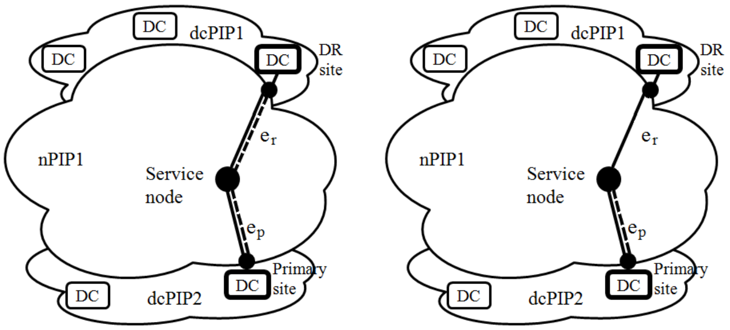
The availability information of the virtual link candidates contains the available bandwidth if bandwidth constraints are applied. Similarly, the amount of node resources can be shared in presence of node resource constraints. The connectivity information of the virtual link candidates is sufficient for building the virtual network and routing the services. If the services need to be transmitted to/from a certain location, this can be ensured by specifying the node location while requesting virtual nodes. In our example optimization related information of the virtual links is the cost and the delay information. The cost of each virtual network resource candidate should be specified by the PIP to the VNO. Additionally, end-to-end delay information of each virtual link can be made available to the VNO without giving the actual physical mapping. In the simulations, we considered only the propagation delay of the virtual links, however, the model can be directly applied for end-to-end delay calculation if this information is available from the PIP. Finally, the disjointness information of the virtual link candidates should be given to the VNO such that it can provision resilience in its virtual network. This can be realized by defining SRLGs containing the virtual link candidates sharing the same failure risk. If e.g. physical edge disjointness is requested, all virtual links sharing the same physical edge are grouped since they would all be affected by the failure of this physical edge. Similarly, SRLGs for node or sub-network disjointness can be formed. Another option is building a set of virtual link pairs, which are not physically disjoint, and providing this information to the VNO as modeled in the MILPs. Using this information the VNOs can then form their virtual networks using the proposed MILPs. As seen from the simulation results the proposed models can be applied in realistic physical topologies for various scenarios. Moreover, even though only the results for NobelUS and NobelEU topologies are presented, we expect the complexity of the problem not to be affected by larger networks if the same number of paths between each node pair are used in the model. Thus, the models are applicable for any kind of physical topologies.

As discussed above, in VNO-Resilience, either the VNO should have knowledge about the disjointness properties of the physical equipment, or this information should be signaled on request from the PIP to the VNO depending on their interface. Similarly, PIP-Resilience might involve communication and information exchange between peer PIPs if the paths need to span multiple PIP domains. In this case, either the PIPs have to coordinate the resilience design among each other, or a third party can be used to combine the resources of the PIPs and lease a resilient virtual network to the VNO.

Another important point is choosing the location of the physical DCs. As mentioned in this section, the distance of the primary and backup sites should be decided according to the fault and disaster types, against which the PIPs want to provide protection. In certain cases placing the servers in different buildings might be sufficient, whereas recovery in case of natural disasters like hurricanes, tsunamis, earthquakes, floods etc. would require larger physical distances and possibly different networks. Moreover, the state synchronization strategy for the primary and DR sites should be decided on according to the service specific needs [64]. Depending on the required fail-over time of the requested services and the physical distances between the two DC sites either shared systems can be used where the load is shared on both servers during normal operation or standby systems where the traffic is redirected to the backup site only in case of failure.

## 6.5 Hybrid Resilience

In this section, we introduce the hybrid resilience models, a mixture of the VNO-Resilience and PIP-Resilience models, where this time the DC resilience is provided by the VNO, and the network resilience is provided by the PIP(s). In the remainder of this section, two hybrid resilience models are introduced and a comprehensive performance evaluation



(a) Hybrid All Paths Protected (HAP): Each service node is connected in the virtual layer to two datacenter sites via resilient virtual links

(b) Hybrid Primary Protected (HPP): Each service node is connected in the virtual layer to two datacenter sites, where the primary path constitutes of resilient virtual links but not the secondary path

Figure 6.10: Proposed resilience models

of all the four resilience alternatives is presented. The first hybrid resilience model, Hybrid All paths Protected (HAP), is a direct application of the basic model with $n_{dc} = 2$ using resilient virtual links. The second one, Hybrid Primary Protected (HPP), has only protection for the path leading to the primary site. The details of the two models are explained in the following.

### 6.5.1 Hybrid All paths Protected (HAP) - All paths protected

In this model, for each cloud service, the service node is connected to $n_{dc}$ DC sites in the virtual layer, where differently than the VNO-Resilience case, these two paths do not need to be disjoint because network resilience is realized by 1:1 protection mapping of the virtual links in the physical layer as shown in Figure 6.10a. Therefore, the basic model is directly applied with the only addition of constraint (6.22) ensuring that the two chosen DC sites are physically in disjoint availability regions and the $n_{dc}$ value is taken as 2 similar to the PIP-Resilience and VNO-Resilience implementations. In this model, using resilient virtual links, all paths leading to the DC sites are protected in the physical layer by the PIP(s).

### 6.5.2 Hybrid Primary Protected (HPP) - Only primary site path protected

In this model, similar to the HAP model, DC resilience is provided by the VNO and network resilience is provided by the PIP(s), with the difference that only the path leading to the primary site is protected in the physical layer as shown in Figure 6.10b. Therefore, if there is a failure in the network affecting the primary path of a service, it is resolved in the physical layer and the service is still routed on the same virtual path to the primary site. If the primary site DC fails, the service is routed in the virtual layer to one of the DR sites. These paths leading to the DR sites are not physically protected since it is assumed that a simultaneous DC and network failure will not happen. As like the former models, only a single DR site is included in the implementation of this model for the simulations.

In order to be able to differentiate among the paths leading to the primary and DR sites, one needs to divide the set of virtual links into two subsets, namely the resilient links, $L_r$ and single path mapped links, $L_n$. The resilient links have the same mapping as in

PIP-Resilience and the single path mapped ones as in VNO-Resilience. The constraints compared with all paths protected model are given in the following.

The constraints (6.8), (6.9) and (6.10) are replaced by (6.26)-(6.27), (6.28)-(6.29) and (6.30), respectively. The new constraints enable the differentiation among the primary and DR sites and hence among the primary and backup paths. The flow conservation constraint for the primary path only uses the resilient virtual links and similarly the one for backup paths only single path mapped links.

$$\sum_{c \in C} a_{s,c} = 1 \quad \forall s \in S \tag{6.26}$$

$$\sum_{c \in C} a'_{s,c} = n_{dc} - 1 \quad \forall s \in S \tag{6.27}$$

$$\begin{gathered} \sum_{l: v \in E_l^r} \beta_{d,l} = \begin{cases} a_{s,c} & \text{if } v = s \text{ or } v = c \\ 2\delta_{d,v} & \text{otherwise} \end{cases} \\ \forall d = (s,c) \in D_u, \; v \in V \end{gathered} \tag{6.28}$$

$$\begin{gathered} \sum_{l: v \in E_l^n} \beta_{d,l} = \begin{cases} a'_{s,c} & \text{if } v = s \text{ or } v = c \\ 2\delta_{d,v} & \text{otherwise} \end{cases} \\ \forall d = (s,c) \in D_u, \; v \in V \end{gathered} \tag{6.29}$$

$$\delta_{d,v} = a_{s,c} + a'_{s,c} \quad \forall d = (s,c) \in D_u, \; v \in \{s, c\} \tag{6.30}$$

Additionally (6.13), (6.16) and (6.19) are replaced by (6.31), (6.32) and (6.33), respectively, due to the distinction introduced among the DC sites .

$$y_c \geq a_{s,c} + a'_{s,c} \quad \forall c \in C, \; s \in S \tag{6.31}$$

$$y_c \leq \sum_{s \in S} (a_{s,c} + a'_{s,c}) \quad \forall c \in C \tag{6.32}$$

$$z_c \geq \sum_{s \in S} (a_{s,c} + a'_{s,c}) \, r_d \quad \forall c \in C \text{ with } d = (s,c) \tag{6.33}$$

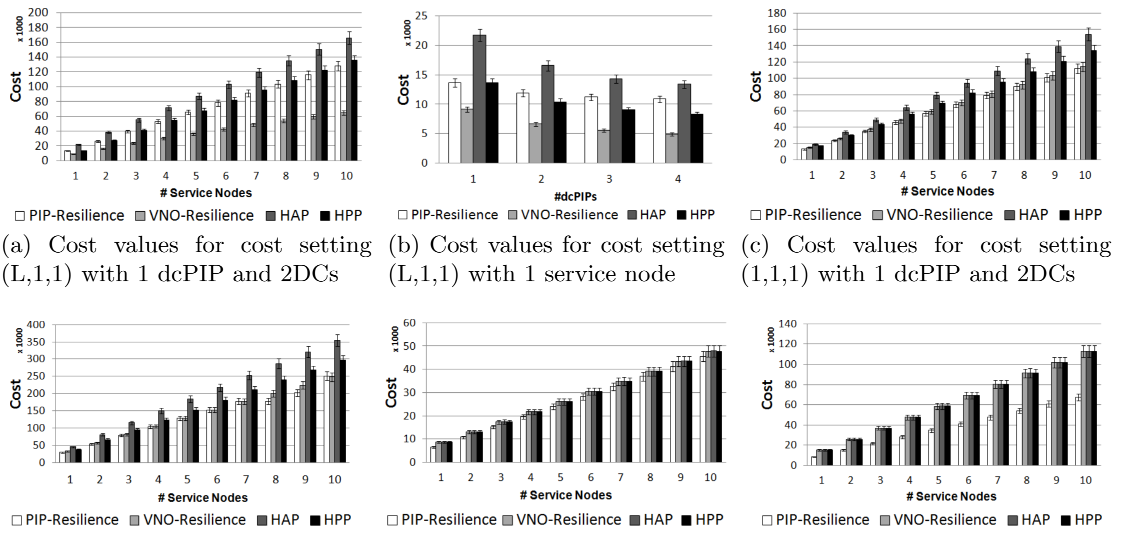
Finally, (6.34) and (6.35) need to be added to the model in order to ensure that all the sites are located in different availability regions. Constraint (6.34) ensures the diversity of the primary and DR sites, and constraint (6.35) ensures the diversity among the DR sites.

$$a_{s,c_1} + a'_{s,c_2} \leq 1 \quad \forall s \in S, \; (c_1, c_2) \in R \tag{6.34}$$

$$a'_{s,c_1} + a'_{s,c_2} \leq 1 \quad \forall s \in S, \; (c_1, c_2) \in R \tag{6.35}$$

### 6.5.3 Performance Analysis of All Resilience Alternatives

In this subsection a detailed comparison of all the four proposed models with different resilience options is presented. This performance evaluation is done in terms of virtual network setup cost, service latency, network resource utilization and virtual network complexity, whose results are provided in the corresponding parts in the following. The same parameter settings are used as in the former section, which are given in Section A.1.4.



(a) Cost values for cost setting (L,1,1) with 1 dcPIP and 2DCs

(b) Cost values for cost setting (L,1,1) with 1 service node

(c) Cost values for cost setting (1,1,1) with 1 dcPIP and 2DCs

(d) Cost values for cost setting (L,A,A) with 1 dcPIP and 2DCs

(e) Cost values for cost setting (1,1,A) with 1 dcPIP and 2DCs

(f) Cost values for cost setting (1,A,1) with 1 dcPIP and 2DCs

Figure 6.11: Virtual network setup cost performance comparison of all the proposed models

#### 6.5.3.1 Virtual Network Setup Cost

Figures 6.11a - 6.11f present the comparison of the absolute cost values of all the introduced four models. These simulations are performed using the NobelEU topology. Figures 6.11a and 6.11b show the cost values for the cost setting (L,1,1). For all the virtual network sizes and different number of dcPIPs, VNO-Resilience results in the lowest cost compared with the other models. For a single dcPIP, the models are listed in the following with an ascending virtual network setup cost as VNO-Resilience, PIP-Resilience, HPP and HAP. The amount of excess cost compared with VNO-Resilience is 50-130%, 50-80% and 140-170% for PIP-Resilience, HPP and HAP, respectively. The high cost observed by HPP and HAP is due to the high number of resilient virtual links involved in these models. With more than one dcPIP, the order of HPP and PIP-Resilience changes, i.e. HPP results in lower virtual network setup cost compared with PIP-Resilience, due to the higher number of DC options available to the HPP model. Finally, increasing the virtual network size increases the absolute cost values and increasing the number of the dcPIPs decreases the absolute virtual network cost values. However, the relative cost gain of VNO-Resilience over the other models increases in both cases.

Using cost setting (1,1,1), the cost gain depends on the resilience premium, whose upper limit can be calculated according to (6.50) and (6.51), which causes the virtual network setup cost for VNO-Resilience and PIP-Resilience to be equal, as discussed in detail in Section 6.7. For cost settings (1,1,1), (1,1,A) and (1,A,1) using cost optimization, the delay is not minimized and takes a random value depending on the selected DC site. Therefore, if there are many available DC sites, cost optimization for these cost settings might result in higher delay values compared with delay optimization. If the service delay is important, delay constraints should be added to the MILP for cost optimization. This results in similar virtual network costs with acceptable delay characteristics.

In the simulations the $r_{\text{PIP}}$ value is taken as 2. Figure 6.11c shows the absolute cost values using the cost setting (1,1,1). The presented values are for a single dcPIP, however the trend in the results does not change with changing dcPIP amount. Therefore, for any dcPIP value it is observed that PIP-Resilience results in the lowest virtual network setup cost while VNO-Resilience, HPP and HAP follow it in the ascending order, where the cost gain of PIP-Resilience over VNO-Resilience, HPP and HAP is 15%, 31% and 46% for one service node, respectively. With an increasing virtual network size, this gain decreases slightly.

Cost optimization using cost setting (L,A,A) results in comparable cost values for PIP-Resilience and VNO-Resilience. PIP-Resilience has slightly lower values of around 0-10%, as shown in Figure 6.11d. Like for the other cost settings, HAP results in the highest virtual network setup cost with a difference of 30% and HPP with a difference of 25% compared with PIP-Resilience. In this cost setting, as in (L,1,1), the physical length of the virtual links is used as the link cost factor and, hence, it also optimizes for the delay implicitly.

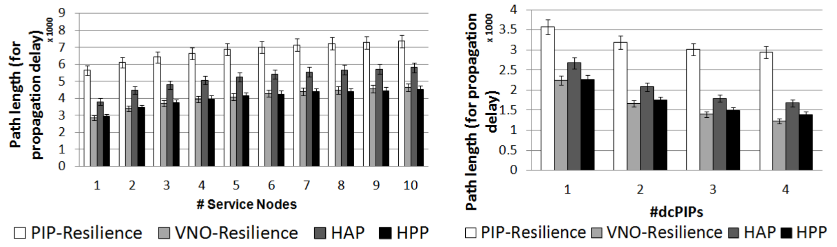
For cost setting (1,1,A) and (1,A,1), VNO-Resilience, HPP and HAP result in almost the same average virtual network setup cost values due to negligible link cost difference compared with the dominant node/VM cost values, while PIP-Resilience results in lower virtual network cost compared with them due to the lower number of VMs and virtual nodes involved in PIP-Resilience. As shown in Figure 6.11e, for cost setting (1,1,A), the cost gain of PIP-Resilience decreases with increasing number of service nodes since it causes the capacity-dependent cost of the VM to be the dominant factor compared with the fixed setup cost. For one service node there is a difference of 33%, which decreases to 5% for 10 service nodes according to our simulations performed with 1 dcPIP and 2 DCs, NobelEU topology and cost optimization.

Using cost setting (1,A,1), where the virtual node cost is the dominant factor, PIP-Resilience always results in cheaper virtual networks compared with the other three models since the number of virtual nodes used in VNO-Resilience, HAP and HPP and their capacity is much higher compared with PIP-Resilience. The cost gain of PIP-Resilience is 75% for a single service node and decreases slightly with increasing virtual network size, reaching 68% for 10 service nodes.

#### 6.5.3.2 Service Latency

In this part the service latency comparison of the proposed models is provided. The presented results in Figure 6.12 are obtained using the delay minimization objective function. Since delay optimization is used, the applied cost setting does not affect the results. Figure 6.12a shows the variation in the maximum delay for increasing number of service nodes. The overall latency increases slightly with increasing number of service nodes, where the trend for the four models remains unchanged. PIP-Resilience has the highest service latency due to the limited number of DC choice and the routing of the services to the DR site over the primary site in the physical layer. The difference of PIP-Resilience compared with VNO-Resilience and HPP, which have very similar results, is around 40%. The results of HAP lie in the middle of PIP-Resilience and VNO-Resilience with a 20% increase compared with the VNO-Resilience due to the usage of resilient links, which increases the delay compared to having path protection within the virtual layer.

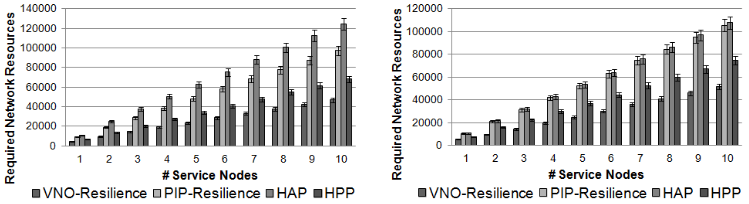
The average delay results are aligned with the maximum delay. Figure 6.12b shows the average service latency performance of the models using delay minimization and varying the number of the dcPIPs for three service nodes. The average delay decreases with increasing number of DC providers for all the models. For PIP-Resilience rather a slight change is observed since increasing the number of the dcPIPs increases only the probability of having a better connected dcPIP with nearer DC locations. For the other models the delay is decreased by 40% using 4 dcPIPs as compared with the case of having a single dcPIP due to the increased options of DCs with each additional provider. For four dcPIPs, PIP-Resilience results in 58% and HAP in 17% higher service latency compared with VNO-Resilience, respectively. HPP and VNO-Resilience have very similar latency performance in accordance with the maximum delay simulation.



(a) Maximum delay performance with 1 dcPIP and 2DCs

(b) Average delay performance with 3 service nodes

Figure 6.12: Service delay performance comparison of all the proposed models - Simulations performed using delay minimization objective function.



(a) Network resource requirement for cost setting (L,1,1) with 1 dcPIP and 2DCs

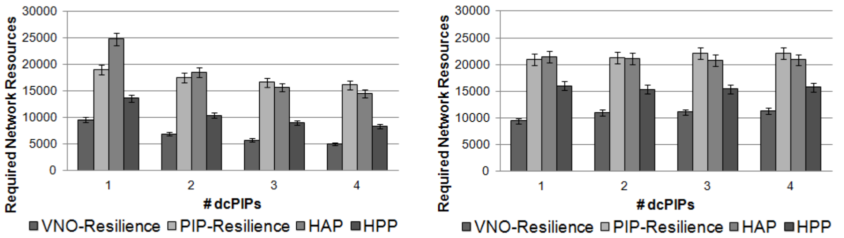
(b) Network resource requirement for cost setting (1,1,1) with 1 dcPIP and 2DCs

Figure 6.13: Network resource requirement comparison of all the proposed models for selected cost settings

#### 6.5.3.3 Network Resource Usage

We present in this part the comparison of the four proposed models in terms of their network resource usage to fulfill the requirements of the same set of cloud services. The results are shown with a varying number of service nodes for one dcPIP with two DCs. The cost settings (L,1,1) and (L,A,A) result in very similar network usage values, where the results for (L,1,1) are shown in Figure 6.13a. VNO-Resilience is the most efficient model in terms of the physical network resources. HPP follows VNO-Resilience with a 40% increase in the amount of required network resources. Finally, PIP-Resilience results in 110% and HAP in 150% higher network resource requirement compared with VNO-Resilience, respectively. In terms of resource requirements the remaining three cost settings, namely the (1,1,1), (1,1,A) and (1,A,1), behave also similarly to each other. The results for (1,1,1) are shown in Figure 6.13b. Compared with Figure 6.13a, all models result in higher average network resource requirement values except for HAP, where this effect is caused due to decreased link cost dominance and hence lesser re-usage of the virtual links as shown in the following part. For these three cost settings, VNO-Resilience is still the most efficient with HPP, PIP-Resilience and HAP following it with around 35%, 100% and 105% resource requirement increase, respectively.

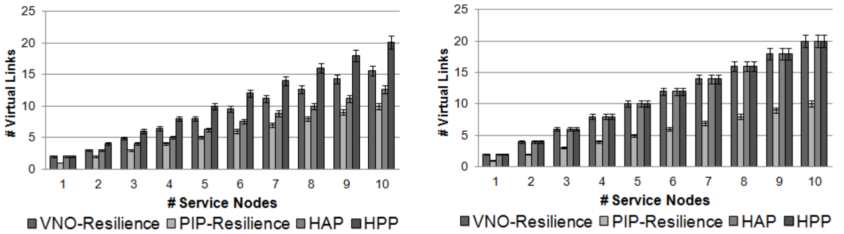
Figure 6.14 provides a network resource comparison of the resilience models under different cost settings and varying number of dcPIPs. It is seen in Figure 6.14b that for the cost settings with fixed link cost values, namely for (1,1,1), (1,1,A) and (1,A,1), the number of the available DCs does not affect the results. However, for the cost settings (L,1,1) and (L,A,A), the total resource requirement decreases for all the models with additional DC availability and HAP model outperforms PIP-Resilience with three dcPIPs available in the network due to the restricted single-domain view of PIP-Resilience.



(a) Network resource requirement for cost setting (L,1,1) with 2 service nodes

(b) Network resource requirement for cost setting (1,1,1) with 2 service nodes

Figure 6.14: Network resource requirement comparison of the models for varying number of dcPIPs



(a) Number of used virtual links for cost setting (L,1,1) with 1 dcPIP and 2DCs

(b) Number of used virtual links for cost setting (1,1,1) with 1 dcPIP and 2DCs

Figure 6.15: Number of used virtual links for all the proposed models for selected cost settings

#### 6.5.3.4 Virtual Network Complexity

Finally, the complexity of the resulting virtual network using the different resilience options is discussed in terms of the required number of the virtual links. Similar to network resource utilization, (L,1,1) and (L,A,A), and (1,1,1), (1,1,A) and (1,A,1) behave very similarly to each other, respectively. The number of used virtual links for (L,1,1) are shown in Figure 6.15a. PIP-Resilience has the lowest number of virtual links due to the physical layer resilience provisioning. It is followed by HAP, VNO-Resilience and HPP with around 20%, 40% and 100% more virtual links, respectively. If the cost of the virtual links are independent of their length, VNO-Resilience, HAP and HPP result in the same virtual link values and have around 100% more virtual links than PIP-Resilience as shown in Figure 6.15b.

## 6.6 Analytical Delay Analysis of the Proposed Models

In this section an analytical comparison of the models in terms of service latency (delay) is presented. The delay performance is analyzed for uniform DC placement, which can be considered as a good approximation given that the PIPs would aim to have coverage in all parts of a network. The worst-case and the best-case delay gain of the VNO-Resilience in comparison with PIP-Resilience is formulated. We also present simulation results which verify the delay analysis.

The average delay considering both normal operation and failure cases for VNO-Resilience is always less than or equal to the average delay of PIP-Resilience because in VNO-Resilience, the services can be routed to any two DCs, and in PIP-Resilience they are routed to one DC and redirected to a second one in case of a DC failure. The amount of this delay gain depends on the number of DCs. The average delay for VNO-Resilience and

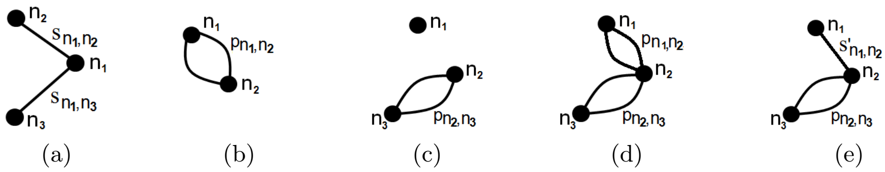


Figure 6.16: DR site selection: (a) VNO-Resilience: both sites selected simultaneously, (b) PIP-Resilience: primary site selected first and then (c) DR site is selected, (d) HAP: Both sites selected simultaneously and connected to service nodes using disjoint path pairs, (e) HPP: Both sites selected simultaneously, where the primary path is resilient and the secondary path is the shortest path

for PIP-Resilience is given as a function of number of DCs in (6.36) and (6.37), respectively. The parameter $q_{dc}$ denotes the total number of DCs available in the network, i.e. all DCs from all dcPIPs, and $q_{dc_i}$ denotes the number of DCs belonging to a PIP $i$. Transmission in fiber is assumed, where the propagation speed is $0.67c$ with $c$ being the speed of light. Furthermore, it is assumed that the DCs are placed randomly on the physical network, the service requesting node is selected randomly and the physical network and the DCs have unlimited capacity. These functions are independent of the number of service nodes, which does not affect the average delay in both models. In this analysis and in the simulations presented at the end of this section, we apply link diversity (an extension to other types of diversity is straightforward).

$$D_{VNO} = \frac{l_V(q_{dc})}{0.67c} \tag{6.36}$$

$$D_{PIP} = \frac{2 \cdot l_{P,P}(q_{dc}) + l_{P,DR}(q_{dc_i} - 1)}{2 \cdot 0.67c} \tag{6.37}$$

$$a_s = \frac{1}{2|N|\binom{|N|-1}{2}} \sum_{n_1 \in N} \sum_{\substack{(n_2,n_3) \in N^2: \\ n_1 \neq n_2 \neq n_3}} (l_{s_{n_1,n_2}} + l_{s_{n_1,n_3}}) \tag{6.38}$$

$$a_p = \frac{1}{2|N|(|N|-1)} \sum_{n_1 \in N} \sum_{n_2 \in N: n_1 \neq n_2} l_{p_{n_1,n_2}} \tag{6.39}$$

$s_{n_1,n_2}$ and $s_{n_1,n_3}$ are shortest disjoint paths from $n_1$ to $n_2$ and $n_3$ as shown in Figure 6.16a. $l_s$ is the length of the path $s$ and $a_s$ is the average length of the disjoint paths calculated over all possible nodes of the physical network for the paths leading from that node to all possible node pairs as given in (6.38). $p_{n_1,n_2}$ is the shortest disjoint path pair between $n_1$ and $n_2$ as shown in Figure 6.16b, and $a_p$ is the average length of all the paths in shortest disjoint path pairs in the physical network as given in (6.39). $l_V(q_{dc})$ is the average length of all working and backup path pairs resulting by VNO-Resilience given as a function of $q_{dc}$. $l_V$ takes the value $a_s$ if there are only two DCs available and takes the value $m_s$ if all other nodes of the physical network are populated with DCs, where $m_s$ is the average of the minimum path lengths for each physical node to two arbitrary DC locations as given in (6.40).

$$m_s = \frac{1}{2|N|} \sum_{n_1 \in N} \min_{\substack{(n_2,n_3) \in N^2: \\ n_1 \neq n_2 \neq n_3}} (l_{s_{n_1,n_2}} + l_{s_{n_1,n_3}}) \tag{6.40}$$

$l_{P,P}(q_{dc})$ is the length of the primary path pair in PIP-Resilience given as a function of $q_{dc}$. For each service node $n_1$ and the selected site, the shortest disjoint path pair is calculated as shown in Figure 6.16b. For a single DC available in the network, namely for $q_{dc} = 1$, we get the $a_p$ as the average length. For $q_{dc} = |N| - 1$, namely if all nodes except the service node are populated with DCs, the function has the value $m_p$, which is the average of the shortest of the disjoint path pair lengths from each node as given in (6.41).

$$m_p = \frac{1}{2|N|} \sum_{n_1 \in N} \min_{n_2 \in N: n_1 \neq n_2} l_{p_{n_1,n_2}} \tag{6.41}$$

Finally, $l_{P,DR}(q_{dc_i})$ is the average length of the paths used to connect the primary site with the DR site given in terms of the remaining DCs for the choice of the DR site in the cloud, from which the primary DC has been selected. For the rest of the calculations, a single dcPIP is assumed, hence $q_{dc_i} = q_{dc}$. The difference to $l_{P,P}$ is that the nodes $n_1$ and $n_2$ are pre-selected while calculating $p_{n_2,n_3}$ as shown in Figure 6.16c. $l_{P,DR}$ takes the value $a_p$ if there is only 1 possible DR site in the cloud. Furthermore, the DR site selection strategy affects the values that this function takes for $q_{dc} \geq 2$. Therefore, we differentiate between $l_{P,DR,R}$ for the random DC selection strategy and $l_{P,DR,S}$ for the shortest delay strategy. The values of $l_{P,DR,S}$ decrease with increasing number of available DCs, $q_{dc}$, since it can optimize its choice over a broader range of DCs. However, the change in the values of $l_{P,DR,R}$ with increasing $q_{dc}$ is negligible for large networks.

If there is only one dcPIP with two DCs in the network, the VNO can only select these DCs. However, in PIP-Resilience the primary DC can be selected among the two DCs to minimize the delay. Therefore, this situation leads to the average worst-case delay gain for the VNO as given in (6.42).

$$Gain_{D,worst} = \frac{\frac{2 \cdot l_{P,P}(2) + a_p}{2} - a_s}{a_s} \tag{6.42}$$

For the considered strategies, the best case for the VNO occurs when the network is fully populated with DCs except for the service node, and when the dcPIPs apply the random DC selection strategy, which causes the DR site selection to be optimized only in VNO-Resilience. The average delay for this case is given in (6.43) for random DC placement.

$$Gain_{D,best} = \frac{\frac{2 \cdot m_p + l_{P,DR,R}(|N|-2)}{2} - m_s}{m_s} \tag{6.43}$$

The functions $l_V$, $l_{P,P}$ and $l_{P,DR}$ depend on the topology of the physical network. For full-mesh topologies with unit edge length, $l_V$ takes the value $s$ and the other two take the value $p$, respectively, independently of the number of the nodes and DCs in the physical network. The value of $p$ and $s$ is always 3 and 2 units, respectively. Therefore, the worst and best case delay gain is equal and is 1.25 in this case. For ring networks with unit edge length, $l_{P,P}$ and $l_{P,DR}$ are again constant and equal to $p$ since the value of $p$ is always equal to the total length of all edges in the network, namely $|N|$ units. However, the value of $l_V$ will be changing depending on $q_{dc}$. It takes the value $a_s$ for $q_{dc} = 2$, which is equal to $\frac{2|N|}{3}$, and $m_s$ for $q_{dc} = |N| - 1$, which is 2 units in length. Therefore, the worst-case delay gain is equal to the full-mesh network case with 1.25 but having all nodes populated with DCs yields a much higher gain depending on the node count of the topology, given as $\frac{3|N|}{4} - 1$. Having arbitrary edge lengths causes all factors of the gain function to be multiplied with the sum of all the edge lengths in the topology, and hence, does not affect the value of the average best case gain. Considering that the ring topology is a special case causing $p$ to be equal to the sum of the edge lengths, it is the case where the highest gain occurs between the two models. Therefore, $\frac{3|N|}{4} - 1$ is the upper bound of the average delay gain for VNO-Resilience for any type of physical topology with $|N|$ nodes.

In case of HAP, both DC sites are chosen simultaneously as in VNO-Resilience with the difference that network resilience is provided by mapping the links connecting the service nodes and the DC sites on two shortest disjoint path pairs, $p_{n_1,n_2}$ and $p_{n_2,n_3}$ as given in Figure 6.16d. The average length of these two shortest disjoint path pairs, $a_{p'}$, for a given network with two random DC locations is calculated as given in (6.44). If the whole network except for the service nodes is populated with DCs, the average length can be minimized by choosing the closest sites and is calculated as given in (6.45).

$$a_{p'} = \frac{1}{4|N|\binom{|N|-1}{2}} \sum_{n_1 \in N} \sum_{\substack{(n_2,n_3)\in N^2: \\ n_1 \neq n_2 \neq n_3}} (l_{p_{n_1,n_2}} + l_{p_{n_1,n_3}}) \tag{6.44}$$

$$m_{p'} = \frac{1}{4|N|\binom{|N|-1}{2}} \sum_{n_1 \in N} \min_{\substack{(n_2,n_3)\in N^2: \\ n_1 \neq n_2 \neq n_3}} (l_{p_{n_1,n_2}} + l_{p_{n_1,n_3}}) \tag{6.45}$$

The difference of HPP to HAP is that in HPP only the primary path, namely the path leading to the primary site, is protected as shown in Figure 6.16e. The secondary path, ${s'}_{n_1,n_2}$, can simply be chosen as the shortest path connecting the service node to the DR site. For any given network, the average length of these two paths, $a_{s'}$, for two random DC locations can be calculated as shown in (6.46). Similar to the HAP case, if the whole network is populated with DCs except the service node, the average length is minimized by choosing the closest sites, and it is calculated as given in (6.47).

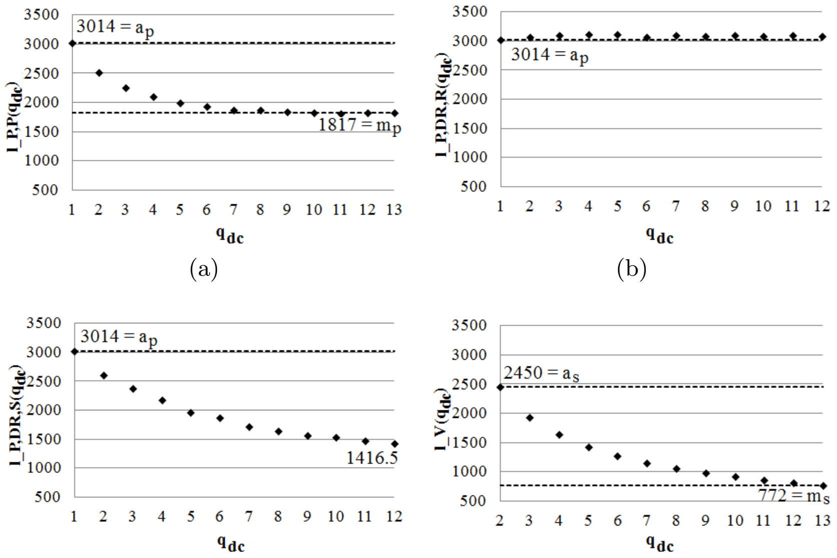
$$a_{s'} = \frac{1}{3|N|\binom{|N|-1}{2}} \sum_{n_1 \in N} \sum_{\substack{(n_2,n_3)\in N^2: \\ n_1 \neq n_2 \neq n_3}} (l_{s'_{n_1,n_2}} + l_{p_{n_1,n_3}}) \tag{6.46}$$

$$m_{s'} = \frac{1}{3|N|\binom{|N|-1}{2}} \sum_{n_1 \in N} \min_{\substack{(n_2,n_3)\in N^2: \\ n_1 \neq n_2 \neq n_3}} (l_{s'_{n_1,n_2}} + l_{p_{n_1,n_3}}) \tag{6.47}$$

Comparing (6.44)-(6.45) with (6.46)-(6.47), one can conclude that HPP always has either equal or better average delay performance compared with HAP because the length of the shortest path $l_{s'_{n_1,n_2}}$ is always equal to or lower than the average length of the disjoint path pair between the same nodes, $\frac{l_{p_{n_1,n_2}}}{2}$. Similarly, since in PIP-Resilience the backup path traverses through the first DC site, it always results in a higher delay compared with both HAP and HPP. Therefore, the order of the observed average delay values of the different models is as follows: PIP-Resilience $>$ HAP $\geq$ HPP. Regarding the comparison of VNO-Resilience with HAP and HPP, the answer is not trivial. However, it can be concluded that the delay performance of VNO-Resilience is better than HAP and in the same range as HPP. Note that these results are in accordance with the simulation results presented in Section 6.5.3.2.

### 6.6.1 Verification of the Delay Analysis via Simulation

In this subsection, the simulation results for the analytical models are shown. We simulated the average delay values resulting in VNO-Resilience and PIP-Resilience using the NobelUS network without availability regions, which validate our mathematical delay models formulated in this section. For a smaller number of DCs, we calculate the results for all possible combinations of DC and service node locations when using one service node. Due to computational complexity, for more than 4 DCs, instead of trying each possible combination, we obtain the results within a ±1% confidence interval at a confidence level of 95% using random DC locations, where for each DC combination all possible service



(a) (b) (c) (d)

Figure 6.17: Simulated values for the length metrics (a) $l_{P,P}$, (b) $l_{P,DR,R}$, (c) $l_{P,DR,S}$ and (d) $l_V$

nodes are simulated. The values of the four functions relating the number of the DCs to the delay occurring in VNO-Resilience and PIP-Resilience are given in Figure 6.17. The values of the functions are calculated by using the equations (6.36)-(6.41) for the simulated values. It is seen that all functions start with either the value of $a_s$ or $a_p$, respectively. $l_{P,P}$ and $l_V$ reach the values 1816 and 768, which are in the $\pm 1\%$ interval of their true values $m_p$ and $m_s$, respectively. For the NobelEU topology, the functions show the same trend where the exact values differ slightly due to their dependence on the topological properties.

## 6.7 Cost and Resilience Premium Analysis

In this section, the behavior of different cost settings is analyzed using different resilience models, and an upper bound for the resilience premium in PIP-Resilience is provided.

Considering the effect of different cost factors, for the cost setting (L,1,1), VNO-Resilience always results in cheaper virtual networks, since the cost of the virtual network is directly related to the physical length of the virtual links, and the link cost is the dominant factor in the virtual network setup cost. This result is aligned with our results above for the delay gain. For the cases, where the virtual node cost or the VM cost is the dominant factor, namely for (1,A,1) and (1,1,A), PIP-Resilience always results in cheaper virtual networks since it results in less number of virtual nodes and VMs compared with VNO-Resilience. However, for cost setting (1,1,A), the difference decreases with increasing number of service nodes because the resilient VM usage becomes the dominant cost compared with the VM setup cost.

Finally, for cost settings (1,1,1), (1,A,1) and (1,1,A), where the link cost has a fixed value, independent of the value of $A$ the minimum cost virtual network has a single DC attached to it in PIP-Resilience and two DCs in VNO-Resilience because having an additional DC site increases the total cost and no delay constraints are applied. Using a minimal number of DCs also minimizes the number of the used nodes. Therefore, the number of the total available DCs and dcPIPs does not effect the results. Moreover, the optimal virtual network has a direct link from each source node to the DC for PIP-Resilience and to the two DCs for VNO-Resilience. Since the fixed cost of a link is equal to its unit capacity cost, sharing the virtual links among different services increases the total capacity dependent cost more than the savings due to the decreased number of virtual links. This topological behavior is independent of the actual values of the link, node and VM costs. For these cases the cost of the virtual network can be calculated for VNO-Resilience and PIP-Resilience as given in (6.48) and (6.49), where $n$ is the number of the service nodes, $u$ is the unit capacity and $r_{\mathrm{PIP}}$ is the resilience premium. It is assumed that 1 unit VM capacity requires 1 unit bandwidth on the links and 1 unit capacity on the network nodes. The capacity requests of all service nodes are assumed to be equal.

$$\varepsilon_{\mathit{VNO,fixed}} = 2n(1+u)\lambda_l + (n+2+4nu)\mu_v + 2(1+nu)\phi_c \tag{6.48}$$

$$\varepsilon_{\mathit{PIP,fixed}} = (n+2nu)r_{\mathrm{PIP}}\lambda_l + (n+1+2nu)\mu_v + (1+2nu)\phi_c \tag{6.49}$$

A PIP can adjust its resilience premium $r_{\mathrm{PIP}}$ in these cost settings to achieve a competitive price offer. The upper limit for the $r_{\mathrm{PIP}}$ value given in (6.50) makes the virtual network setup cost equal for VNO-Resilience and PIP-Resilience. Using (1,1,1), the limit of the $r_{\mathrm{PIP}}$ value is larger than 2 and with increasing $n$, it approaches 2. Therefore, having $r_{\mathrm{PIP}} = 2$ for (1,1,1) would ensure a PIP to always have a competitive offer. For cost settings (1,A,1) and (1,1,A), the lower limit of $r_{\mathrm{PIP}}$ is again equal to 2 and the higher the node and VM cost, the more independently of the cost considerations can a PIP choose the $r_{\mathrm{PIP}}$ value.

$$r_{\mathrm{PIP}} \leq 1 + \frac{n\lambda_l + (1+2nu)\mu_v + \phi_c}{n(1+2u)\lambda_l} \tag{6.50}$$

For the general case, if $n_{dc} - 1$ DC sites are requested for resilience purposes, the inequality for the resilience premium $r_{\mathrm{PIP}}$ is formulated as in (6.51).

$$r_{\mathrm{PIP}} \leq \frac{n_{dc}n(1+u)\lambda_l + (n_{dc}-1)(1+2nu)\mu_v + (n_{dc}-1)\phi_c}{n(1+2u)\lambda_l} \tag{6.51}$$

Both models of Hybrid Resilience, namely the case when all paths are protected (HAP) and when only the primary path is protected (HPP) behave similar to VNO-Resilience with cost optimization using the fixed cost settings. For both cases, the resulting virtual network uses only $n_{dc}$ DCs and includes a virtual link from each service node to each DC, where $n_{dc} - 1 : 1$ redundancy is applied. In case of HAP, all the $n_{dc}$ paths are protected, which results in an excess cost compared to VNO-Resilience increasing linearly with increasing number of service nodes $n$ as given in (6.52).

$$\varepsilon_{\mathit{HAP,fixed}} = \varepsilon_{\mathit{VNO,fixed}} + n_{dc}nr\lambda_l \tag{6.52}$$

For the case of HPP, only the primary path is protected, which results in an additional cost compared to VNO-Resilience due to the resilience premium applied on this path as given in (6.53).

$$\varepsilon_{\mathit{HAP,fixed}} = \varepsilon_{\mathit{VNO,fixed}} + nr\lambda_l \tag{6.53}$$

In conclusion, both HAP and HPP always result in higher virtual network setup cost compared with VNO-Resilience for the fixed cost settings and the difference increases linearly with increasing virtual network size.

## 6.8 Heuristic Algorithms for Virtual Network Design with Cloud Services

A general framework of heuristics for virtual network design is presented in Section 5.6. The heuristics for virtual network design with cloud services are based on that heuristic framework and the differences compared with the presented framework will be discussed in this section.

The general logic behind the proposed heuristics realizing resilient virtual network design for cloud services is based on the existing heuristics for shared protection in the physical layer [137, 138] similar to the unicast case, and anycast routing heuristics [61]. In the existing heuristics, first a subset of DC sites are selected and then the services are routed from the service source nodes to the DC sites iteratively. During the routing process, a cost value is assigned to each edge. Since sharing of the redundant resources is desired, the cost of an edge, which is already used in the protection path of another service, is set to zero if the working path of that service is disjoint with the current one, which is the prerequisite of shared protection. This will force the algorithm to choose preferably such edges, which maximize the shared redundant capacity.

In our case, we want to design a resilient virtual network topology for cloud services, where a VNO needs to pay a certain fee to the PIP(s) for the rental of the virtual resources and the setup of the corresponding virtual network using the selected resources. This fee or in other words the virtual network setup cost consists of the fixed setup and capacity dependent cost components as introduced in the former sections. The general case for virtual network design and service routing resembles the shared protection heuristics in the physical layer. When a link, node or VM is not used yet, for a new service routing both cost components have to be counted. If it is already used by another service, the fixed cost of this resource is discarded. The main difference of the cloud service heuristics with Section 5.6 is hence the addition of the DC selection part prior to each routing. This mechanism is shortly described in the following. For each service, firstly, a certain number of minimum cost primary DC candidates are selected according to the total cost of the DC and its connection path to the source node of the service. Then, for each of these primary DCs, a list of possible DR sites is calculated, where the number of the primary and corresponding DR sites is a variable of the algorithm. Afterwards, the minimum cost DR site and its connection path are selected among the list of the DR site candidates. In Chapter 7, the details of the heuristic algorithm are explained and two examples are shown for the shared protection use case, whose performance evaluation is also presented.

## 6.9 Summary

In a world where the businesses and private applications are increasingly dependent on cloud solutions and where network virtualization is seen as a key enabler for future networks, it is highly important to design end-to-end resilient virtual networks for cloud services. In this chapter, we address this problem by introducing novel optimization models formulated as MILPs enabling a cost-efficient resilient virtual network design for both the communication network and cloud domains. Our models offer resilience either in the virtual or physical layer or use a combination of them. Moreover, we evaluate the performance of these models and the cloud connection models for existing virtual networks to provide answers to the question of how much benefit virtual layer resilience offers. Afterwards, we present a latency, cost and resilience premium analysis for the models and finally introduce the general heuristic framework for calculating resilient virtual networks for cloud services.

The following research questions from Section 1.3 are answered in this chapter:

**Q1.1:** Does the prior art provide answers to the resilient virtual network design problem? If not, where are the shortcomings?

Similar to the case with connectivity services, in virtual network design for cloud services, the existing literature falls short in providing solutions to this problem. It is either only focused on a sub-problem like mapping the virtual network or routing the services and takes the second part as pre-given like in the case of virtual network embedding and overlay networks, or totally discards the network virtualization aspects as e.g. in the physical layer anycast routing and multi-layer resilience areas.

**Q1.3:** How can resilient virtual network design be extended to cover cloud resources in order to provide end-to-end resilience for cloud services?

There are two fundamental alternatives for realizing this, namely by providing resilience for the cloud connection either in the virtual or in the physical layer. In the former, the virtual network is attached directly to a primary site and to one or more DR sites. The paths leading to these sites need to be physically disjoint. In case of a failure, the traffic is redirected within the virtual network to the DR site. In case of physical layer resilience, from the VNO perspective the virtual network is connected to a single DC site. In case of failure, the traffic is re-routed in the physical layer from the primary site to the DR site within the domain of the cloud provider. An evaluation of these models is also provided in this chapter.

**Q1.4:** How can resilient virtual networks be designed to serve end-to-end resilient cloud service requests?

To answer this question, MILP models are presented in this chapter, which provide virtual or physical layer resilience or make use of a combination of those, namely the hybrid resilience. All models allow simultaneous optimization of service routing within the virtual network, DC site selection for the services and mapping of the virtual links and nodes. We provide the details of the models and evaluate their performance in comparison with prior approaches. Our simulations show that in around 50% of the simulation runs SPM, which uses a fixed shortest path mapping for the virtual links, fails to find a resilient virtual network solution, while this value is only 0.02% for the proposed models. If using additional nodes is allowed for SPM which are neither source nor destination nodes of the services, namely using SPMwAN, the simulations show that on average the resulting virtual network topology includes a higher number of virtual links and nodes compared with the proposed models, where the difference is around 45% for the number of virtual links, and 40% for the number of virtual nodes. Hence, allowing additional nodes enables the prior approach to find resilient network solutions. This however increases the setup and maintenance costs of the virtual network significantly. Finally, comparing the proposed approach, which allows a combined optimization of IT and network domains, with separate optimization, we show that the maximum service latency, which can be guaranteed inside the virtual network, can be reduced to half.

**Q1.5:** To cope with the possible scalability problems of the virtual network design models, what kind of heuristics can be used for resilient virtual network design?

In this chapter, the heuristic framework from Chapter 5 is extended with the selection of the DC sites together with their connection paths to enable the design of resilient virtual networks for cloud services. The methodology behind that is briefly described in this chapter. More details are given then in Chapter 7 for the special use case of shared protection and their applicability and performance are evaluated.

**Q2.2:** Does virtual layer resilience bring any benefits in terms of virtual network setup cost, service latency, physical resource utilization and complexity?

This chapter first aims to provide an initial analysis for the selection of the resilience layer for cloud connections, namely the virtual or physical layer resilience. Two scenarios are evaluated where in the first one the same virtual network is used for both models and in the second one optimal resilient virtual design models are utilized. Both scenarios show a clear benefit of using virtual layer resilience for cloud connections. Our quantitative study shows that the latency gain of virtual layer resilience compared with having it in the physical layer is about 60% if the same virtual network is used for both models and reaches 120% for virtual network topologies designed with the corresponding resilience options under certain circumstances. Using the end-to-end optimization models, the maximum guaranteed latency gain of virtual layer resilience reaches 210% for the same settings.

Comparing the performance of the four optimization models, which provide resilience in the virtual or in the physical layer for both network and IT domains, or delegate the network resilience to the physical layer while providing the DC resilience in the virtual layer, we observe different behaviors using different cost settings. In terms of virtual network setup cost, VNO-Resilience outperforms the others in case of the dominance of the link cost. For other cost settings, PIP-Resilience provides a lower cost, where in case of similar link, node and VM costs, the difference vanishes. If fixed cost values are used, the cost performance ratio depends on the value of the resilience premium. A detailed analysis of the resilience premium selection strategy is also presented in this chapter. The hybrid models always result in the highest cost values, where for the cases of node or VM cost dominance their difference with VNO-Resilience results vanishes.

In terms of latency, PIP-Resilience performs the worst due to limited DC choice and the routing of the services to the DR site over the primary site in the physical layer. HAP follows PIP-Resilience and VNO-Resilience and HPP perform similar to each other and have the lowest latency results.

Similar to latency results, in terms of physical network utilization, VNO-Resilience performs best, which is followed in the ascending order of resource requirements by the HPP, PIP-Resilience and HAP. The performance difference between the models increases when the virtual link cost depends on its physical length. The reason of high resource requirement of PIP-Resilience is the link-level resilience provisioning and for HAP the increased level of resilience.

Finally, in terms of virtual network complexity PIP-Resilience offers the best solution followed by HAP, VNO-Resilience and HPP. Using fixed cost values, the last three result in the same number of virtual links.

All in all, virtual layer resilience shows the highest benefits in terms of latency and network utilization and can offer some gain in terms of cost depending on the used cost setting. However, it results in higher virtual network complexity. Hybrid resilience models show a weak cost performance but a high latency gain similar to VNO-Resilience. In cases, where a VNO wants to delegate the network resilience to the PIP(s) due to e.g. lack of the necessary knowledge of network operation, they can provide feasible solutions.

## 6.10 Statement on Author's Contributions

The Section 6.2.1 of this chapter is based on [83], and Section 6.2.2 is based on part of [88] introducing the cloud connection models and their performance evaluation. Sections 6.3 and 6.4 are an extended version of [124] and [125]. In all of these publications the design of the proposed models and evaluations have been carried out by the author. In the thesis a more detailed explanation of the used inputs and the outputs of the MILP is presented, and network utilization and complexity results are added to the performance evaluation section. Finally, a cost and resilience premium analysis and a heuristic framework are provided.

# Part IV

# Enhanced Virtual Network Design Models

# 7. Shared Protection in Virtual Networks

Resilience is a key feature of today's networks and clouds, and its importance will increase with increasing dependency of businesses on these solutions. However, resilience results in a high cost for both business roles due to redundant resource requirements. To cope with this problem, in today's networks shared protection is applied. Resilience mechanisms can be divided into two groups, namely restoration and protection as shown in Figure 7.1. Restoration mechanisms are flexible, however they result in a relatively long recovery time due to the signaling and path calculation processes. Therefore, if instantaneous recovery is needed, protection mechanisms are preferred. Protection mechanisms are again divided into two groups as dedicated and shared protection. In dedicated protection, each working path of a service has a dedicated protection path, which is used in case of a failure. Therefore, dedicated protection can be quite costly, requiring at least the double capacity compared with non-resilient routing. As a result, as mentioned before, shared protection mechanisms can be preferred, which offer a fast recovery as well as lower capacity requirements compared with dedicated protection.

In this chapter, we apply the idea of shared protection to virtual networks by allowing sharing of redundant *virtual* network and IT resources between different services. Sharing of redundant virtualized IT resources for DR can be realized by keeping a backup of the primary site VMs and re-instantiating them on the shared idle resources of the DR site in case of failure or even before a natural disaster like a hurricane hits the area to avoid service interruption. The redundant virtual link resources can be also shared if the working paths of the services using the same redundant link are disjoint. An example is shown in

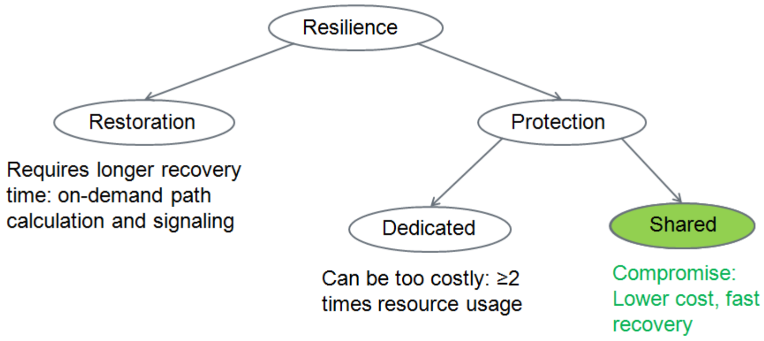

Resilience
Restoration
Protection
Requires longer recovery time: on-demand path calculation and signaling
Dedicated
Shared
Can be too costly: ≥2 times resource usage
Compromise: Lower cost, fast recovery

Figure 7.1: Classification of resilience mechanisms with their advantages and disadvantages

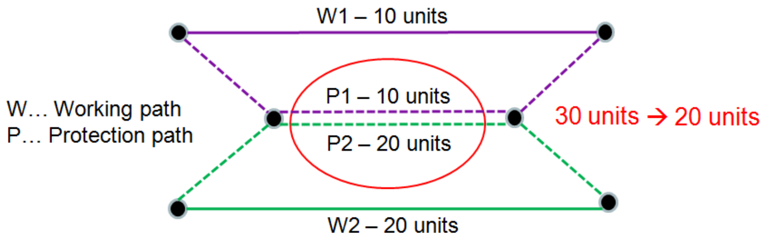


Figure 7.2: Shared protection example: If the working paths are disjoint, the capacity on the protection paths can be shared. Without sharing 30 units capacity is required on the common link due to the protection paths P1 and P2 carrying 10 and 20 units capacity, respectively. If the working paths W1 and W2 are disjoint, i.e. if these paths don't fail simultaneously, the capacity on this link can be shared, and hence, only 20 units capacity is sufficient, which is the maximum capacity of the two paths.

Figure 7.2, where the required capacity on the common redundant link can be reduced from 30 units to 20 units if the working paths are disjoint. Note that it is not sufficient to have virtual layer disjointness. These paths have to be physically disjoint.

Recalling the aims of the business roles, shared protection in virtual networks offers a win-win situation by reducing the price of the virtual network for the VNOs and enhancing the physical resource efficiency for the PIPs. In this chapter, we introduce architecture extensions and information exchange enabling the application of the shared protection concept on virtual networks. Afterwards, we introduce the optimization models and heuristics with shared protection both for connectivity and for cloud services. For each of them the performance of the models and algorithms is evaluated to determine the advantages of applying shared protection compared with dedicated protection. Moreover, the choice of the layer where shared protection should be applied is also discussed.

Section 7.2 of this chapter and the results presented in Section 7.3.1.3 are based on our publication [139]. Moreover, Section 7.4 is an extended version of [140].

## 7.1 Related Work and Contributions

Shared protection stands out as a feasible resilience mechanism, which makes use of the fast recovery of protection mechanisms and lowers the cost compared with dedicated protection, which is one of the main concerns of operators. As discussed in Chapter 9, operators usually observe a trade-off between an increased level of resilience provisioning and increased cost. In this aspect, shared protection offers a good solution for the current and future operators.

Shared protection is widely studied for current network technologies like WDM networks to evaluate its advantages in terms of network capacity and cost [141, 142]. Like the general resilience mechanisms, shared protection can be divided to certain groups depending on the scope and type of protection. One grouping depends on the part of the connection, which is protected. The protection and hence sharing of the redundant resources can be made path-based, where an end-to-end working path is protected via a protection path [143, 144] or it can be link or segment-based [145, 146], where link or segment protection can be seen as similar to failure-dependent protection in the sense that which detour to take also depends on which link has failed [147]. The other option is having failure-independent shared protection, where the protection paths are designed to be disjoint from the working path, are chosen to maximize the redundant resource sharing and serve for all failure cases [148]. Finally, the sharing can be done demand-wise, meaning that the protection capacity of different protection paths of the same demand can be shared [149, 150], or the protection paths of different disjoint demands can be shared. The former has the advantage of being simpler in terms of network management, however it requires the usage of multiple disjoint paths for a demand, which might not be feasible in sparse networks. The latter works also for the 1+1 protection case, where there is one protection path for one working path. In our work we use the latter and our shared protection models use - as in the models introduced in the other chapters - failure-independent path protection.

In this chapter, we apply the concept of shared protection, which has been extensively studied for physical networks as described above, to network virtualization. Looking to the available literature in this area, the authors in [151] focus on PIP-layer shared protection and assume the virtual network topology to be given, and hence, do not do sharing of the virtual resources in the sense that we aim to do. In [152], the authors focus on DC resource sharing and only allow the sharing of the bandwidth for the same service between different backup locations, which is not applicable if a single backup site is used. Moreover, both of these works discard the virtual network setup cost aspect, which is a crucial part in the PIP-VNO business relationship. In this chapter, we first introduce the necessary architecture extensions, which are needed to be able to have shared protection in virtual networks. Afterwards, we investigate both the resource utilization and cost savings enabled by shared protection in virtual networks for connectivity and cloud services and discuss at which layer it is better to apply sharing in terms of cost, resource utilization, service latency and complexity metrics.

## 7.2 Architecture Framework for Shared Protection

In this section, first, a general architecture for dynamic creation of virtual networks is presented. Afterwards, it is shown how this architecture needs to be extended to allow the sharing of redundant resources among different virtual networks or services, i.e. allowing application of shared protection in virtual networks. To keep the architecture as general as possible, the VNP business role, which is a broker between the different PIP and VNO domains, is also considered in this architecture. For the remainder of the chapter, however, this business role is considered to be included to the VNO to maintain simplicity.

### 7.2.1 Dynamic Creation of Virtual Networks

Before the application of shared protection for virtual networks, as a first step, we describe the setup of virtual networks as an implementation example with amended GMPLS techniques. The reason for this choice is that it is expected that vendors and operators are highly interested in saving on recent capital investments in their GMPLS network infrastructure when introducing the support of network virtualization. However, this description serves only as an example, and the concepts and models presented in this thesis are technology independent.

Several architectures and procedures are discussed in [153] and [37] for the creation of virtual networks. Our model is based on the model given in [75] and further detailed such that with minor modification virtual networks can be dynamically set up and operated, as e.g. in SDN architectures. The dynamic creation of virtual networks based on the roles SP, VNO, VNP and PIP are split into three steps. These are; 1) dissemination of virtual resources from the PIP to the VNP (e.g. via an augmented OSPF), which allows the VNP to have a limited view of the virtual resources possibly available at the PIP, 2) reservation of the virtual network by VNO via VNP at the PIP, and 3) handing over the control of the resulting virtual network from the PIP via the VNP to the VNO for its further operation. The virtual resource request can be expressed by an augmented Resource reSerVation Protocol Traffic Engineering (RSVP-TE) protocol such that after having calculated the mapping of the resources to the physical layer, the RSVP-TE Path message carries the Virtual networkID1 (distinguishing the individual virtual network from any other) and the particular virtual resource within a new sub-object as additions to the Explicit Route Object (ERO) and Secondary Explicit Route Object (SERO) objects, as defined in RFC4875 [154]. While traversing the hosting physical nodes the augmented RSVP-TE process attaches the selected virtual resources and the corresponding virtual links to the local nodal physical interconnections and external forwarding plane links.

As with the conventional RSVP-TE procedure in the Resv message, the indication of successful allocation of the virtual resources is returned back. In particular, the Resv message is used to carry the addresses of the virtual resources knowing that finally the VNO needs to have access to the resources exclusively being granted. We suggest that the addresses for the related configuration interfaces of the virtual resources are to be collected along the paths in a new sub-object attached to an augmented Record Route Object (RRO) and Secondary Record Route Object (SRRO) contained in the RSVP-TE Resv message. Finally, the Resv message arrives at the VNO enabling the VNO to configure the resources.

Above we mostly focused on the mechanism to set up a network for virtualized resources for plain connectivity between the given endpoints. However, in future carrier grade communication networks more than this simple connectivity is required. As of today, any customer and operator expects to be able to utilize and provide services with certain requirements, as for instance contracted via the SLAs. Especially in cases where the VNP requires composing a network across several PIPs, the guaranteed service availability needs to receive special attention. The VNO requests dynamically a highly reliable virtual network from the VNP. Then, the VNP either performs resilience on its own responsibility or delegates it down to the PIPs. By whoever and wherever resilience is provisioned, a dynamic mechanism is needed between the roles to decide and signal who is responsible for a particular virtual network. As RFC4872 [155] defines the protection object which indicates what kind of protection is to be applied (unprotected, 1+1, 1:N, etc.). This can be exploited in the scope of network virtualization. With the signaling of the protection object, the VNO requests protection from the VNP/PIP or does not if this object is omitted. In the latter case, the VNP/PIP is not requested to protect the corresponding virtual network and the VNO can create its own solution for protection. On the other hand, the VNO does not know the physical topology and the physical location of its virtual resources. None of the roles is able to further optimize because of the limited knowledge within their own realm. Therefore, in the next sections we look at if and how one can gain from a shared protection in a virtualized environment.

### 7.2.2 Shared Protection Architecture for Virtual Networks

Network virtualization allows a PIP to host multiple virtual networks sharing its physical resources. We extend this concept by allowing the VNO (or VNP) to apply the same concept as the PIP with the difference that this time the virtual resources are shared among different virtual networks.

The main difference compared with conventional networks is that a VNO cannot simply share the resources since it has a lack of knowledge about the physical topology and the physical location of the virtual resources. Hence, there is a need for a certain information exchange between the roles. For that purpose we introduce a new atomic activity called SHARE. This activity takes as an input parameter the resources of different virtual networks, which should be shared. It is a sharing request to be sent from the VNO to the VNP, which can translate it to the PIPs. The PIP evaluates the SHARE request, the results are returned to the VNP, which processes them and reports to the VNO. The required information exchange can be found in Figure 7.3.

A VNO that is the owner of two virtual networks, which are named VNetID1 and VNetID2, may know that these networks may have complementary time requirements in terms of

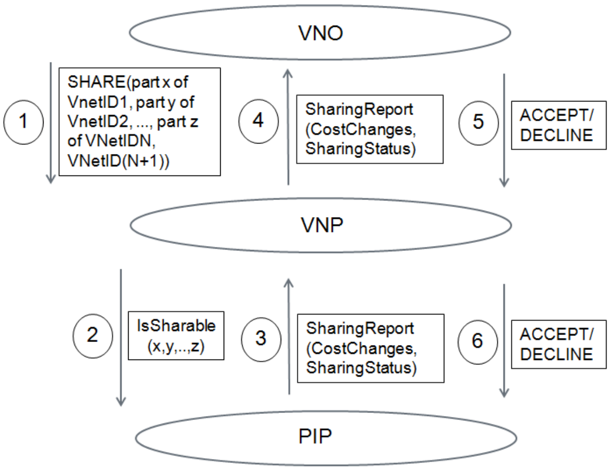

VNO
1
SHARE(part x of VnetID1, part y of VnetID2, ..., part z of VNetIDN, VNetID(N+1))
4
SharingReport (CostChanges, SharingStatus)
5
ACCEPT/ DECLINE
VNP
2
IsSharable (x,y,..,z)
3
SharingReport (CostChanges, SharingStatus)
6
ACCEPT/ DECLINE
PIP

Figure 7.3: Information exchange between the roles

statistical multiplexing for particular parts of the networks or it may wish to protect parts of the networks on a shared underlying physical or virtual network. The driver for this are cost savings for the concerned VNO, the VNP and the PIP in the business relationship. In order to enable this, we propose the following procedure as an implementation example:

**1)** The VNO requests the VNP to check whether it is feasible to share (and at which prices) the indicated (parts of the) virtual networks as signaled in the SHARE message towards the VNP. If the prices are known beforehand (e.g. due to pre-arranged bilateral agreements), the VNO may simply request the VNP to perform the sharing, without any charging negotiation.

If the VNO wants to share particular resources, it expresses itself by requesting from the VNP that the resources of the VNetID1 or a subset of them are to be shared with the resources of the VNetID2 or a subset of them. For instance, the VNO may issue a message, such as the following, in a generalized format towards the VNP: SHARE ((part $x$ of VNetID1, part $y$ of VNetID2), [VNetID3]) by which it is requested to share part $x$ of VNetID1 and part $y$ of VNetID2 on the VNetID3. [ ] denotes that the ID of the VNetID3 might optionally be added in case the requester wants to share the resources on the particular virtual network VNetID3. Of course, a VNO can also request the sharing of a part within a single virtual network.

**2)** On receipt of the message from the VNO, the VNP can reject the request. If not and since the VNP per definition does not know whether the selected resources can be shared on the physical layer, it consults the involved PIP(s) in order to check whether the resources can be shared, and at which prices. It is to be noted that for that sharing there is a win win situation for the involved roles, because the VNO saves money and the PIP can reach a higher utilization rate.

**3)** Eventually the PIP will receive the SHARE message from the VNP. From a functional view it is now up to the PIP to create the new VNetID3 on the real physical substrate owned by the PIP. It is clear that only resources, parts of the VNetID1, which are congruent with VNetID2 can be shared on a new virtual network. Therefore after the receipt of the message from the VNP, the PIP may reject the request. If not, the PIP can continue to perform the sharing by issuing an RSVP-TE Path message on the control plane by which the new virtual resources in VNetID3 are allocated on the physical resources. In the next step, the virtual resources (of VNetID1 and 2) are to be mapped to VNetID3, such that the resources of VNetID1 and VNetID2 are still separated, but completely confined within

VNetID3. This can be achieved by applying the so called label stacking mechanism in the conventional MPLS networks.

Hierarchical LSPs are created by embedding a new client LSP onto an existing server LSP or a new client LSP on a newly created lower level server LSP. However, for the sharing procedure, VNetID1 and VNetID2 already exist but now need to be stacked on VNetID3. As such, the nesting of virtual networks on top of another virtual network is performed by an enhancement of the concept of hierarchical LSPs (also known as LSP nesting: RFC4726 [156] and RFC4206 [157]) by sending a RSVP-TE Path message for each of the LSPs of VNetID1 and VNetID2 to be shared on the nesting LSPs of VNetID3.

This Path message carries a new flag called "sharing" in the existing RSVP-TE object "Attribute Flags for LSP Attributes Object" together with the VNetID1, the RRO, SRRO indicating the particular part of the virtual network to be shared and the VNetID2 (RRO, SRRO) to be nested on VNetID3. In this way, the control plane of the physical resource recognizes that the selected virtual resources of the corresponding virtual networks are to be shared on VNetID3, stacking the labels of each of the virtual networks on top of the label of VNetID3.

As the RSVP-TE Path message traverses along the paths of VNetID3, each physical node is configured accordingly. Once an edge node is reached, it will return the indication of successful configuration back within the RSVP-TE Resv message, which is forwarded to the VNP. Alternatively instead of immediately enforcing the sharing, the PIP may determine the related costs for the reconfiguration and return the result back to the VNP.

**4)** On receipt at the VNP, the VNP may change the related costs as it may also want to participate in the efficiency gain before forwarding the message to the VNO. If the sharing was already performed by the PIP, the message indicating the successful configuration is forwarded to the VNO.

**5)** If the sharing of the resources was already performed, the procedure is completed with the receipt of the message at the VNO. However, if the VNO received an indication about the resulting costs, it may accept or reject the offer by sending a message to the VNP.

**6)** On receipt of the message at the VNP indicating the acceptance or the rejection of the related costs for sharing, the message is forwarded to the PIP and the PIP completes the procedure by finally enforcing the sharing or by omission of the sharing as indicated by the response from the VNP.

It is important to note that as part of this procedure each role in general can object or accept the request from the previous role according to their respective technical or business needs. It is to be noted that the above description mainly consists of two conceptual procedures. One is the case where the involved partners have a price pre-arrangement for the invocation of the service, which shortens the procedure to steps 1 to 4. Another case is the approach where the partners may renegotiate the price for the service of sharing. Then the procedure is completed with step 6.

In case of shared protection, when the resources, which are intended to be shared, are reserved for resilience purposes, there is an additional information exchange required to determine which virtual protection paths are allowed to share resources. Only protection paths belonging to disjoint working paths can share resources. If two virtual working paths share physical links or nodes, it means that they belong to the same SRLG, and hence, they can fail simultaneously. Thus, their protection paths might be required to be used at the same time and resource sharing among them is not possible.

In case the disjointness information is not available, a VNO can send a IS_DISJOINT($R_1$, $R_2$, disjointnessType) message to the VNP, where $R_1$ and $R_2$ are two virtual resources of any type, i.e. they can be of type virtual link, virtual node or even a virtual IT resource - a VM. The disjointnessType defines what type of disjointness is requested. For a virtual link this can be physical link, physical node or sub-network disjointness. Similarly for the node and IT resources it can be physical node/server or sub-network disjointness. The sub-network disjointness can be agreed on between the roles by defining availability regions inside the physical topology.

## 7.3 Sharing of Redundant Virtual Link Resources

### 7.3.1 Optimization Models with Shared Protection

In this section, the MILPs used for optimal virtual network design with shared protection are introduced. We propose two models, one having resilience in the virtual layer, VNO-Resilience and the second one in the physical layer, PIP-Resilience. The models describe the sharing of redundant virtual resources for different services within a virtual network, however, it can be also applied for sharing redundant resources among different virtual networks by defining the services representing separate virtual networks. All the sets, parameters and variables used in both models are given in the following.

- *Sets:*
  - $V$: Set of all the virtual node candidates
  - $L$: Set of all the virtual link candidates
  - $D$: Set of the requested services
  - $E_l$: Set of the endpoints of link $l \in L$
  - $E$: Set of the physical links in the physical network topology
  - $P_l$: Set of the physical links $e \in E$, on which the virtual link $l \in L$ is mapped
  - $W_l$: Set of the physical links $e \in E$, on which the virtual working link $l \in L$ is mapped
  - $B_l$: Set of the physical links $e \in E$, on which the virtual protection (backup) link $l \in L$ is mapped
  - $Z$: Set of virtual link pairs $(j, k) \in L^2$, which share at least one physical edge
- *Parameters:*
  - $b_d$: Requested bandwidth for the service $d \in D$
  - $n_d$: Requested node resources for the service $d \in D$
  - $t_l$: Physical length of link $l \in L$
  - $s_e$: Length of physical edge $e \in E$
  - $\lambda_l$: Fixed setup cost for having a new link $l \in L$
  - $\theta_l$: Setup cost per unit capacity for link $l \in L$
  - $\mu_v$: Fixed setup cost for having a new node $v \in V$
  - $\eta_v$: Setup cost per unit capacity for node $v \in V$
  - $r_{PIP}$: Cost factor of providing PIP-Resilience for a virtual link $l \in L$
  - $r_{NU}$: Cost factor of the network utilization in the cost objective function
- *Variables:*
  - $\beta_{i,d,l}$: Binary variable taking the value of 1 if the link $l \in L$ is used for the $i^{th}$ route of the demand $d \in D$, 0 otherwise
  - $\delta_{i,d,v}$: Binary variable taking the value of 1 if the node $v \in V$ is used for the $i^{th}$ route of the demand $d \in D$, 0 otherwise
  - $\tau_{d,l,l'}$: Binary variable taking the value of 1 if the link $l' \in L$ is used for the protection of the link $l \in L$ for the demand $d \in D$, 0 otherwise
  - $\gamma_l$: Binary variable taking the value of 1 if the link $l \in L$ is in the resulting virtual network, 0 otherwise
  - $\alpha_v$: Binary variable taking the value of 1 if the node $v \in V$ is in the resulting virtual network, 0 otherwise

- $u_l$: Used capacity on link $l \in L$, $u_l \in [0, \infty]$
- $\phi_{l,l'}$: Used capacity on link $l' \in L$, that is used for the protection of the link $l \in L$, $\phi_{l,l'} \in [0, \infty]$
- $\phi_l$: Used protection capacity on link $l \in L$, $\phi_l \in [0, \infty]$
- $\psi_l$: Used working capacity on link $l \in L$, $\psi_l \in [0, \infty]$
- $\omega_v$: Used capacity on node $v \in V$, $\omega_v \in [0, \infty]$
- $\epsilon_{j,d,e}$: Binary variable taking the value of 1 if the physical edge $e \in E$ is used for the $j^{th}$ route of the demand $d \in D$ on the physical substrate, 0 otherwise
- $\nu_{d,e,e'}$: Binary variable taking the value of 1 if the physical edge $e' \in E$ is used for the protection of the physical edge $e \in E$ for the demand $d \in D$ on the physical substrate, 0 otherwise
- $\pi_{e,e'}$: Used capacity on physical edge $e' \in E$ that is used for the protection of the physical edge $e \in E$, $\pi_{e,e'} \in [0, \infty]$
- $\pi_e$: Used protection capacity on physical edge $e \in E$, $\pi_e \in [0, \infty]$
- $\rho_e$: Used working capacity on physical edge $e \in E$, $\rho_e \in [0, \infty]$
- $y_e$: Used capacity on physical edge $e \in E$, $y_e \in [0, \infty]$

#### 7.3.1.1 VNO-Resilience

In VNO-Resilience, resilience is provisioned in the virtual layer. Therefore, shared protection is also applied in the virtual layer by sharing the redundant virtual link resources among different services. This can be only done if the working paths of these services are physically disjoint. This is enabled by providing the physical disjointness information of the virtual links to the VNO as described in Section 7.2.2. Each service is routed on $k$ physically disjoint virtual paths in the virtual network. For simplicity, $k$ is taken as 2 in this model. In the following the constraints used in VNO-Resilience model are explained in detail.

Equation (7.1) is the non-splittable flows conservation constraint and (7.2) states that if a node is used as the source or destination of a service, it would be flagged as used by that service. Constraints (7.3) and (7.4) state that a virtual link or node is part of the resulting virtual network if it is used by any service, respectively. Inequality (7.5) is the node capacity constraint calculating the required capacity on the virtual nodes.

$$\sum_{l \in L : v \in E_l} \beta_{i,d,l} = \begin{cases} 1 & \text{if } v = s \text{ or } v = t \\ 2\delta_{i,d,v} & \text{otherwise} \end{cases} \qquad \forall d = (s,t) \in D,\ v \in V,\ i \in \{1,..,k\} \tag{7.1}$$

$$\delta_{i,d,v} = 1 \quad \forall d = (s,t) \in D,\ v \in (s,t),\ i \in \{1,..,k\} \tag{7.2}$$

$$\gamma_l \geq \beta_{i,d,l} \quad \forall l \in L,\ d \in D,\ i \in \{1,..,k\} \tag{7.3}$$

$$\alpha_v \geq \delta_{i,d,v} \quad \forall v \in V,\ d \in D,\ i \in \{1,..,k\} \tag{7.4}$$

$$\omega_v \geq \sum_{i \in \{1,..,k\}} \sum_{d \in D} \delta_{i,d,v}\, n_d \quad \forall v \in V \tag{7.5}$$

In VNO-Resilience the working and protection paths of each service need to be physically disjoint, which is ensured by (7.6).

$$\beta_{1,d,i} + \beta_{2,d,j} \leq 1 \quad \forall d \in D,\ (i,j) \in Z \tag{7.6}$$

The following constraints enable shared protection in virtual networks. The first path with the path indicator $i = 1$ is assumed to be the working path and the second path with $i = 2$ to be the protection path of a service. It is common to have the working path shorter than or equal to the protection path to decrease the latency of the services in normal operation. Therefore, (7.7) ensures that the first path is shorter than the second one.

$$\sum_{l \in L} t_l \left( \beta_{1,d,l} - \beta_{2,d,l} \right) \leq 0 \quad \forall d \in D \tag{7.7}$$

Constraints (7.8) and (7.9) are used to determine the value of the indicator $\tau_{d,l,l'}$ showing if a certain link $l'$ is used to protect the link $l$ for the service $d$.

$$\beta_{1,d,l} + \beta_{2,d,l'} \leq 1 + \tau_{d,l,l'} \quad \forall d \in D,\ l, l' \in L,\ l \neq l' \tag{7.8}$$

$$2\tau_{d,l,l'} \leq \beta_{1,d,l} + \beta_{2,d,l'} \quad \forall d \in D,\ l, l' \in L,\ l \neq l' \tag{7.9}$$

Equation (7.10) is used to calculate the protection capacity required on a link $l'$ due to the link $l$ with all the involved services. Constraint (7.11) is similar to shared protection in physical networks where the capacity on the protection link $l'$ needs to be at least as much as the maximum requirement coming from the working links using this link. The main difference of shared protection in virtual networks to physical networks is that the working links are not necessarily disjoint and there is the need to check their mappings. Constraint (7.12) calculates the required capacity in case of the physical edge failures, which cause multiple working links of a protection link to fail. In this case, the capacity on the protection link should be at least the sum of the capacities of the affected working links.

$$\phi_{l,l'} = \sum_{d \in D} b_d \, \tau_{d,l,l'} \quad \forall l, l' \in L,\ l \neq l' \tag{7.10}$$

$$\phi_{l'} \geq \phi_{l,l'} \quad \forall l, l' \in L,\ l \neq l' \tag{7.11}$$

$$\phi_{l'} \geq \sum_{e \in E : l \in P_l} \phi_{l,l'} \quad \forall l, l' \in L,\ l \neq l',\ e \notin P_{l'} \tag{7.12}$$

Constraint (7.13) is used to calculate the working capacity on each virtual link as the summation of the requested capacities of all services using that link in their working path. Finally, the total required capacity on each link is calculated in (7.14) as the summation of the required working and protection capacities.

$$\psi_l \geq \sum_{d \in D} \beta_{1,d,l} \, b_d \quad \forall l \in L \tag{7.13}$$

$$u_l = \phi_l + \psi_l \quad \forall l \in L \tag{7.14}$$

The objective function used in VNO-Resilience model minimizing the virtual network setup cost is presented in (7.15). As defined in Chapter 5, the cost consists of the summation of link costs and node costs. Both link and node costs consist of two parts, namely the fixed setup cost of establishing a new virtual link or node and the capacity-dependent cost depending on the required capacity on this link or node.

$$\min \quad \Big( \sum_{l \in L} (\lambda_l \gamma_l + \theta_l u_l) + \sum_{v \in V} (\mu_v \alpha_v + \eta_v \omega_v) \Big) \tag{7.15}$$

#### 7.3.1.2 PIP-Resilience

In case of PIP-Resilience, the resilience is provisioned at the physical layer, and hence, redundant resource sharing is also done in the physical network. In PIP-Resilience, the services are routed on a single path in the virtual network and protection is provided by mapping the virtual links on two disjoint physical paths. The constraints (7.1)–(7.5) form also the basis for PIP-Resilience. Since sharing is performed in the physical network, the capacity calculation of the virtual links is done by simply summing over the capacity requirements of the services using that link as given in (7.16).

$$u_l \geq \sum_{i \in \{1,..,r\}} \sum_{d \in D} \beta_{i,d,l}\, b_d \quad \forall l \in L \tag{7.16}$$

To relate the routing information to the physical mapping, Equations (7.17) and (7.18) set the values of the indicator for the working and protection physical edges depending on the service routing in the virtual network and the mapping of the virtual links.

$$\epsilon_{1,d,e} \geq \beta_{1,d,l} \quad \forall e \in W_l,\ d \in D,\ l \in L \tag{7.17}$$

$$\epsilon_{2,d,e} \geq \beta_{1,d,l} \quad \forall e \in B_l,\ d \in D,\ l \in L \tag{7.18}$$

Similar to the VNO-Resilience model, constraints (7.19) and (7.20) are used for determining the value of the protection edge indicator $\nu_{d,e,e'}$. This variable indicates if an edge $e'$ is used for protection of edge $e$ for a service $d$.

$$\epsilon_{1,d,e} + \epsilon_{2,d,e'} \leq 1 + \nu_{d,e,e'} \quad \forall d \in D,\ e, e' \in E,\ e \neq e' \tag{7.19}$$

$$2\nu_{d,e,e'} \leq \epsilon_{1,d,e} + \epsilon_{2,d,e'} \quad \forall d \in D,\ e, e' \in E,\ e \neq e' \tag{7.20}$$

Constraints (7.21) and (7.22) calculate the protection capacities on the edges per working edge and in total, respectively. Constraint (7.23) is used to calculate the working capacity on the edges and finally, (7.24) computes the total capacity needed on the physical edges.

$$\pi_{e,e'} = \sum_{d \in D} b_d\, \nu_{d,e,e'} \quad \forall e, e' \in E,\ e \neq e' \tag{7.21}$$

$$\pi_{e'} \geq \pi_{e,e'} \quad \forall e, e' \in E,\ e \neq e' \tag{7.22}$$

$$\rho_e \geq \sum_{d \in D} \epsilon_{1,d,e}\, b_d \quad \forall e \in E \tag{7.23}$$

$$y_e = \pi_e + \rho_e \quad \forall e \in E \tag{7.24}$$

The objective function needs to be updated for PIP-Resilience as given in (7.25) with the addition of network utilization minimization and the insertion of the resilience premium compared with (7.15). Physical network utilization should be explicitly minimized in case of PIP-Resilience to optimize sharing of the redundant resources since shared protection in the physical layer does not directly affect the cost of the virtual network. Network utilization is defined as the multiplication of the used capacity on an edge $e$, $y_e$, with the length of that physical edge, $s_e$.

$$\begin{aligned} \min \quad & \Big( \sum_{l \in L} \left( (\lambda_l \gamma_l + \theta_l u_l)\, r_{PIP} \right) + \sum_{e \in E} (y_e s_e)\, r_{NU} \\ & + \sum_{v \in V} (\mu_v \alpha_v + \eta_v \omega_v) \Big) \end{aligned} \tag{7.25}$$

We looked at different cost settings in our analysis, where the cost of the virtual links can be either a fix value or dependent on the physical length of the virtual link as introduced in Chapter 5. In PIP-Resilience, this length is calculated as the sum of the lengths of the two disjoint paths on which the virtual link is mapped. Hence, if the cost of the virtual link is a function of its length, resilience premium due to the resilience provisioning by the PIP is implicitly included to the cost. However, for the case of fixed link cost, we use the resilience premium $r_{PIP}$, which is taken as 2 for our evaluations in this chapter. Finally, we define the weighting factor $r_{NU}$ to balance the weight of the cost and network utilization minimization.

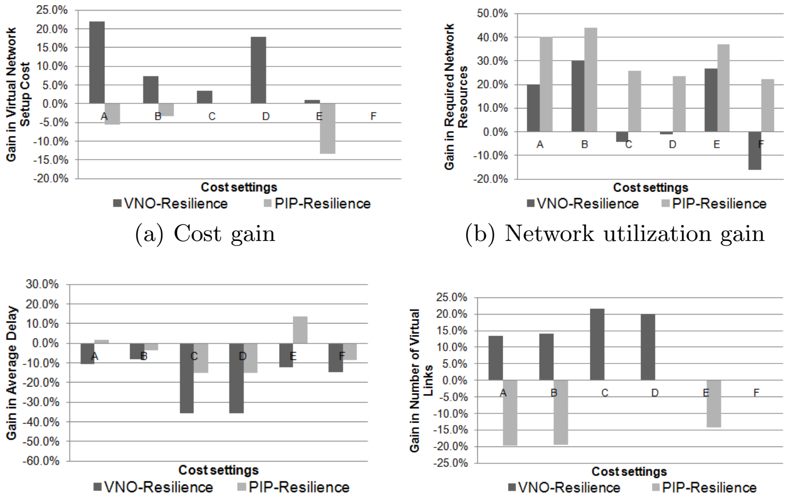
#### 7.3.1.3 Results and Evaluation

In this section the simulation results using the optimization models are presented and evaluated. The simulations are carried out for 3 and 4 virtual service nodes due to the scalability issues. However, they give a good insight for the gain offered by shared protection already for small virtual networks and form a good basis for development of heuristics.

For the simulations, the same setup is used as in Chapter 5. The used cost settings are defined in Table 5.1 and referred in this section in the figures as A=(L,L,1,1), B=(L,L,A,A), C=(1,1,1,1), D=(1,100,1,1), E=(100,1,1,1) and F=(1,1,100,100). Moreover, the value of the weighting factor $r_{NU}$ is calculated as the ratio of the average cost values for each cost setting to the network utilization value. The value of $r_{NU}$ used in the simulations for each cost setting is given as: $r_{NU_A} = 1$, $r_{NU_B} = 2$, $r_{NU_C} = 0.00067$, $r_{NU_D} = 0.02$, $r_{NU_E} = 0.04$ and $r_{NU_F} = 0.067$.

We provide first performance comparisons among the dedicated and shared protection models of VNO-Resilience and PIP-Resilience to evaluate the gain provided by shared protection. We as well compare the performance of the two resilience models for shared protection with different cost settings to give a first insight for the question of at which business role resilience should be provisioned. All the results shown in this section are for three virtual nodes. For the results shown in Figure 7.4, the gain is calculated as the relative percentage difference of the performance values coming from shared and dedicated protection, where the bars show the gain for shared protection. Figure 7.4a shows the gain in virtual network setup cost with shared protection. For the cost settings A and D, where the capacity dependent cost of the virtual links constitute a high portion of the total cost, a cost saving of around 20% is observed for VNO-Resilience. This reduction is caused by the reduced capacity usage with the sharing of the redundant links. For PIP-Resilience the cost remains in the same range since the sharing is performed in the physical layer, which has no direct impact on the virtual network cost. For PIP-Resilience in cost setting E, an increase of virtual network setup cost is observed, which is due to the increased network utilization efficiency that is realized by using more and slightly longer virtual links as depicted in Figures 7.4b, 7.4c and 7.4d, respectively. This increase reaches 23% for 4 virtual nodes. For the remaining cost settings both models provide similar gain levels for 3 and 4 virtual nodes.

Figure 7.4b presents the gain in physical network utilization, which is defined as the multiplication of the used capacity and the length for all the physical edges. Since minimization of the used capacity is directly part of the objective function in case of PIP-Resilience, it provides a high gain of more than 20% for all the cost settings, reaching 44% for cost setting B. VNO-Resilience also provides a gain of 20-30% in network utilization for the cost settings A, B and E, where the link cost is the dominant factor within the total virtual network cost. For cost setting F, a slight increase in the network utilization and delay is observed, which is due to the nature of this cost setting where the link cost has almost no effect in the objective function and hence sharing is not optimized. The network utilization



(a) Cost gain

(b) Network utilization gain

(c) Average delay gain

(d) Gain in number of virtual links

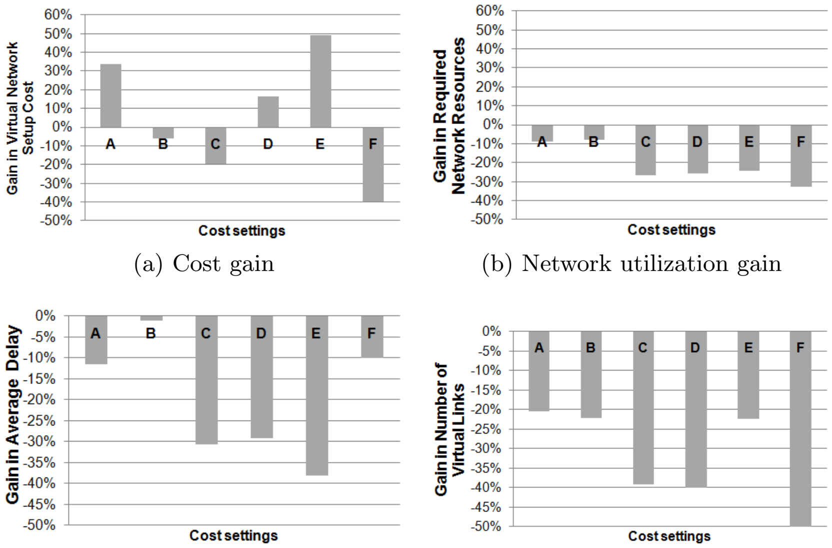
Figure 7.4: Performance of Shared over Dedicated Protection: $(\text{value}_{\text{dedicated}} - \text{value}_{\text{shared}})/\text{value}_{\text{shared}}$

gain increases with increasing number of virtual nodes and reaches 27% and 58% for cost setting A with VNO-Resilience and PIP-Resilience, respectively, 42% for cost setting F and 70% for cost setting E with PIP-Resilience for 4 virtual nodes.

Delay gain of shared over dedicated protection is presented in Figure 7.4c. In the simulations we only considered the propagation delay which is linearly related to the physical length of the path of a service. However, the model can be directly applied for the end-to-end service delay. We observed that shared protection increases the service delay for VNO-Resilience since the number and/or the length of the virtual link mappings is increased to enable sharing. For PIP-Resilience the increase in delay is lower and cost setting E results even in a gain in service delay due to the increased number of virtual links as shown in Figure 7.4d. The results for maximum delay observed inside the virtual network show the same trend as the average delay. For 4 virtual nodes, the delay gain of PIP-Resilience decreases due to the increased network utilization efficiency. However, the difference in delay for VNO-Resilience remains in the same range except for cost settings A and E, where for the former the excess delay reaches 21% and for the latter vanishes completely.

Finally, Figure 7.4d shows the difference in number of virtual links for VNO-Resilience and PIP-Resilience in case of shared vs. dedicated protection. In VNO-Resilience, except for the cost settings E and F, the number of the virtual links is decreased by 10-20%. In cost setting E, already for dedicated protection only 3 virtual links are used since the fixed setup cost of the virtual links is the dominant cost factor and hence it cannot be reduced more. For cost setting F, the link cost is not the dominant factor and hence we don't observe any change in the number of the virtual links. In case of PIP-Resilience, on the opposite to VNO-Resilience, 15-20% more virtual links are used in case of shared protection in cost setting A, B and E, which increase the cost slightly however provides a high network utilization gain.

In conclusion, shared protection reduces the virtual network setup cost for VNO-Resilience up to 22% and the physical network utilization for both VNO-Resilience and PIP-Resilience, whereas for PIP-Resilience this gain can go up to 70% for 4-nodes virtual networks. This gain occurs with the trade-off of increased delay for both models and increased number of virtual links for PIP-Resilience. Finally, in the cost setting F the node cost is dominant



(a) Cost gain

(b) Network utilization gain

(c) Average delay gain

(d) Gain in number virtual networks

Figure 7.5: Performance of VNO-Resilience over PIP-Resilience with shared protection: $(\text{value}_{\text{PIP-Resilience}} - \text{value}_{\text{VNO-Resilience}})/\text{value}_{\text{VNO-Resilience}}$

and hence as it can be observed from the results, it does not bring much benefit when shared protection is applied. Due to that reason for the performance evaluation of the heuristics this cost setting is omitted.

In Figure 7.5 the performance of VNO-Resilience and PIP-Resilience is compared. The results in Figure 7.5 are for 3 virtual nodes and the gain of VNO-Resilience is shown with positive bars. Figure 7.5a shows the cost gain of VNO-Resilience to PIP-Resilience. It is observed that VNO-Resilience results in lower cost when the setup cost of the links is the dominant factor, which allows optimization of sharing and hence cost reduction in VNO-Resilience. The cost gain increases with 4 virtual nodes to 52% for cost setting A and to 83% for cost setting E. In cost settings C and F, where the cost of a node is equal to the link or is the dominant factor, PIP-Resilience offers a lower cost. Hence, the decision of the resilience provisioning layer would depend on the actual cost factors.

Regarding network utilization, delay and number of virtual links, PIP-Resilience always offers better results compared with VNO-Resilience, where the gain of PIP-Resilience can reach 30%, 35% and 50% for network utilization, delay and number of virtual links, respectively, as shown in Figures 7.5b, 7.5c and 7.5d.

As mentioned in the beginning of this subsection, the MILP is not scalable and can be only used for small virtual networks. To overcome this problem, heuristic algorithms are presented in the following chapter and their performance is also evaluated.

### 7.3.2 Heuristics with Shared Protection

In this section, heuristics enabling shared protection in virtual networks are presented in detail. A general framework has been given in Chapter 5. These heuristics are a special case of dedicated protection heuristics, which can be easily obtained from the described algorithms by omitting the sharing part.

The general logic behind the proposed heuristics for virtual networks is based on the existing heuristics for shared protection in the physical layer [137, 138]. In the existing heuristics, the services are routed iteratively and during the routing process, a cost value is assigned to each edge. Since sharing of the redundant resources is desired, the cost of an edge, which is already used in the protection path of another service and if the working path of this service is disjoint with the current one, which is the prerequisite of shared protection, is set to zero. This will push the algorithm to choose such edges, which maximize the shared redundant capacity.

In our case, we want to design a resilient virtual network topology, where a VNO needs to pay a certain fee to the PIP(s) for the rental of the virtual resources and the setup of the corresponding virtual network. The cost of a virtual link or node consists of two parts: The fixed cost for the setup and the capacity dependent cost, which increases with increasing requested capacity on this resource. The virtual network setup cost is given in (7.15).

The general case for virtual network design and service routing resembles the shared protection heuristics in the physical layer. When a link or node not used yet for any service, for a new service routing both cost components have to be considered. If it is already used by another service, the fixed cost of this resource is discarded. Moreover, for shared protection mechanism to be added to the virtual network design, the cost calculation needs to be adapted accordingly. In that case, if a link is already used and is used for the protection path of a service $d_1 \in D$ and the working path of a new service $d_2$ is disjoint with the working path of $d_1$, the cost of this link is reduced to zero. The node capacity is not shared since we only have protection against single physical (and implicitly virtual) link failures. In the following, two heuristic algorithms realizing shared protection in virtual networks are introduced, namely the HillClimber and kBest algorithms.

#### 7.3.2.1 HillClimber Algorithm

For the HillClimber heuristic, as the name implies, always the local best solution is selected. The general problem with the local search algorithms is that they can get stuck at a local optimum. One option is running the algorithm several times starting from different points. However, as the number of the services increases, this method is not scalable. Therefore, we sort the services in the beginning according to the length of their shortest path mappings. Our simulations have shown a combination of the ascending and descending order services provides the best results for our problem.

The details of the HillClimber algorithm are provided in Algorithms 1 - 4. In Algorithm 1, each service is routed once consecutively on the minimum cost path in a Greedy manner. After each routing, this information is saved for the current service and the virtual link usage information is updated as given in Algorithm 2. At the end of the first iteration, the virtual network setup cost is calculated and saved. Then, for each service, the routing of this service is removed, link and node costs are corrected accordingly and the service is re-routed. The virtual network cost is re-calculated with the new routing. If the new cost is lower than the old one, the new routing is kept. Otherwise, the new routing is omitted and the old routing is re-applied. If the old and new costs are equal, again the old routing is kept. If in one complete iteration the routings of all the services remain unchanged or the maximum number of iterations, $i_{max}$ is reached, the algorithm returns.

Two versions of this algorithm exist: one for VNO-Resilience and one for PIP-Resilience. In the former, resilience provisioning is in the virtual layer, and hence, the redundant capacity on the virtual links is shared among different services. In the latter, each service has a single path routing within the virtual network and resilience is provisioned in the physical layer, by mapping each virtual link on two disjoint physical paths. In this case, the physical edge capacity of the protection paths of the virtual links is shared among different virtual links.

**Algorithm 1 :** Shared Protection HillClimber Heuristic for VNO-Resilience.

```
Sort the demands $d \in D$ according to their physical shortest path distance
Set $i = 0$ {Iteration counter}
while $i < i_{\max}$ do
  if $i = 0$ then
    for all $d \in D$ do
      $(p_w, p_p) \leftarrow$ lowest_cost_disjoint_paths($d$,Virtualnetwork)
      set_properties($d$, $(p_w, p_p)$)
    end for
    Calculate $cost_{\text{Virtualnetwork}}$
  else
    $changed \leftarrow$ false
    for all $d \in D$ do
      Remove the routing of $d$ and update the virtual link costs
      $(p_w, p_p) \leftarrow$ lowest_cost_disjoint_paths($d$,Virtualnetwork)
      Calculate $cost_{\text{Virtualnetwork,new}}$
      if $cost_{\text{Virtualnetwork,new}} < cost_{\text{Virtualnetwork}}$ then
        $cost_{\text{Virtualnetwork}} \leftarrow cost_{\text{Virtualnetwork,new}}$
        $changed \leftarrow$ true
        set_properties($d$, $(p_w, p_p)$)
      else
        Re-apply the old routing of $d$ and update the virtual link costs
      end if
    end for
    if $changed = 0$ then
      break
    end if
  end if
end while
```

**Algorithm 2 :** set_properties($d$, $(p_w, p_p)$) - Sets the usage information and the bandwidth usage of links used in the working and protection paths $p_w$ and $p_p$.

```
Set the paths $p_w$ and $p_p$ as the working and protection paths of $d$
for all $l \in p_w$ do
  $w_l = 1$
  $b_{\text{l,working}} \leftarrow b_{\text{l,working}} + b_d$
end for
for all $l \in p_p$ do
  $p_l = 1$
  for all $e \in P'_l : l' \in w$ do
    $i_e, i_l \leftarrow$ index of $e$ and $l$
    $M[i_l][i_e] \leftarrow M[i_l][i_e] + b_d$
  end for
  $b_{\text{l,protection}} \leftarrow \max(M[i_l])$
end for
```

### VNO-Resilience

In VNO-Resilience, as described in the previous section, resilience is provisioned in the virtual layer and shared protection is also applied inside the virtual network. Each service is routed on 2 physically disjoint virtual paths in the virtual network offering 1:1 protection. This can be easily generalized to $k$:1 protection by repeating the disjoint path calculation $k$ times for a given working path and set of former backup paths.

In VNO-Resilience as the cost function directly the virtual network setup cost is used as given in (7.15). The core of the algorithm is given in Algorithm 3, where the lowest cost disjoint paths routing of a service $d$ is calculated. Firstly, $k_w$ lowest cost working paths are calculated using the $k_w$-lowest cost paths algorithm, where the current cost of each virtual link depends on its current usage. Afterwards, for each working path a lowest cost disjoint backup path is calculated. While doing this, the cost of the virtual links are updated according to the corresponding working path. Note that, sharing of the redundant resources on a link $l$ among different services is allowed only if the working paths of these services are physically disjoint. Virtual link disjointness is not sufficient. In case some virtual links of the working paths share a physical edge, for the protection capacity calculation the sum of their capacities need to be considered. This is enabled by using a matrix $M$, which holds the required protection capacity on each virtual link due to each physical edge, on which a virtual link is mapped that is used as part of a working path. Each time a new service is routed, this matrix is updated. Finally, from all the working and backup path pairs, the pair with the lowest cost is returned for the service $d$.

**Algorithm 3 :** lowest_cost_disjoint_paths($d$,Virtualnetwork) - Calculates the lowest cost disjoint path pair for a demand $d$.

```
$cost_{\text{current}} \leftarrow \infty$
$W \leftarrow k_w$ lowest cost virtual working paths of demand $d$
for all $w \in W$ do
    Set the link usage flags for the working path $w$
    for all $l \in L$ do
        if $w_l = 0$ and $p_l = 0$ then
            $cost_l \leftarrow \lambda_l + \theta_l b_d$
        else if $w_l = 1$ and $p_l = 0$ then
            $cost_l \leftarrow \theta_l b_d$
        else
            $b_{\text{l,protection,new}} \leftarrow b_{\text{l,protection}}$
            for all $e \in P'_l : l' \in w$ do
                $i_e \leftarrow$ index of $e$
                $i_l \leftarrow$ index of $l$
                if $M[i_l][i_e] + b_d > b_{\text{l,protection,new}}$ then
                    $b_{\text{l,protection,new}} \leftarrow M[i_l][i_e] + b_d$
                end if
            end for
            $cost_l \leftarrow \theta_l(b_{\text{l,protection,new}} - b_{\text{l,protection}})$
        end if
    end for
    Determine the lowest-cost path $p$ that is disjoint from the working path $w$
    Calculate the cost $cost(w,p)$ of having the paths $w$ and $p$ in the virtual network
    if $cost(w,p) < cost_{\text{current}}$ then
        $p_w \leftarrow w$
        $p_p \leftarrow p$
        $cost_{\text{current}} \leftarrow cost(w,p)$
    end if
    Reset the link usage flags (remove the routing of $w$)
end for
return Paths $p_w$ and $p_p$
```

### PIP-Resilience

In case of PIP-Resilience, resilience is provisioned by the physical layer, and hence, sharing is also done within the physical network. The services are routed on a single path in the virtual network, and protection against single physical link failures is provided by mapping each virtual link on two disjoint physical paths. In PIP-Resilience, if we only use the virtual network cost in the cost function, physical resource sharing cannot be enforced since it is transparent to the virtual layer. Therefore, the cost function of PIP-Resilience is updated by adding network utilization cost as given in (7.25), where $r_{NU}$ is the weighting factor for this cost component. Network utilization is defined as the multiplication of the used capacity on a physical edge $e$, $y_e$, with its length, $s_e$. Moreover, a resilience premium, $r_{PIP}$ is added to the link cost since in PIP-Resilience resilient virtual links are provided, and its value is taken as 2 for this section.

Since resilience is provisioned in the physical layer, in Algorithm 1 instead of two virtual paths only a single virtual path is calculated per service, where each virtual link is now mapped on two disjoint physical paths. Therefore, Algorithm 3 is changed with Algorithm

4. The paths are first calculated according to their virtual link costs, then network utilization is calculated for each path and the path's total cost is updated. The lowest cost path is chosen according to the total cost value. Moreover, since disjointness is directly applied in the physical layer, the matrix $M$ is changed with $M'$, which holds the protection capacity information of each physical edge $e$ per each physical edge $e'$, where $e$ is part of the protection path and $e'$ is part of the working path of virtual links.

The complexity of both VNO-Resilience and PIP-Resilience versions of the HillClimber algorithm depends on the number of the working paths calculated, $k_w$, the number of the demands, $|D|$, and the number of the iterations $i$. The used $k_w$-lowest cost paths algorithm is a variant of the Bellman-Ford algorithm but instead of storing the best path, it stores the $k_w$ best paths at each pass. Therefore, it yields a complexity of O($k_w|L||V|$), where $|L|$ is the number of the virtual link candidates and $|V|$ is the number of the virtual nodes. Worst case complexity is hence O($i_{max}k_w|D||L||V|$).

**Algorithm 4 :** lowest_cost_path($d$,Virtualnetwork) - Calculates the lowest cost path for the demand $d$.

```
$cost_{\text{current}} \leftarrow \infty$
for all $l \in L$ do
    if $w_l = 0$ then
        $cost_l \leftarrow \lambda_l + \theta_l b_d$
    else
        $cost_l \leftarrow \theta_l b_d$
    end if
end for
$W \leftarrow k_w$ lowest virtual network cost paths for $d$
for all $w \in W$ do
    $cost_{\text{path}} \leftarrow$ Cost of the virtual path $w$
    $cost_{\text{networkUsage}} \leftarrow 0$
    for all $l \in w$ do
        for $e \in W_l$ do
            $cost_{\text{networkUsage}} \leftarrow cost_{\text{networkUsage}} + t_e b_d$
        end for
        for all $e \in B_l$ do
            $b_{\text{e,protection,new}} \leftarrow b_{\text{e,protection}}$
            for all $e' \in W_l$ do
                $i_e, i_{e'} \leftarrow$ index of $e$ and $e'$
                if $M'[i_e][i_{e'}] + b_d > b_{\text{e,protection,new}}$ then
                    $b_{\text{e,protection,new}} \leftarrow M'[i_e][i_{e'}] + b_d$
                end if
            end for
            $cost_{\text{networkUsage}} \leftarrow cost_{\text{networkUsage}} + t_l(b_{\text{e,protection,new}} - b_{\text{e,protection}})$
        end for
    end for
    $cost_{\text{total}} \leftarrow cost_{\text{path}} + r_{\text{NU}} cost_{\text{networkUsage}}$
    if $cost_{\text{total}} < cost_{\text{current}}$ then
        $cost_{\text{current}} \leftarrow cost_{\text{total}}$
        $p_w \leftarrow w$
    end if
end for
return $p_w$
```

#### 7.3.2.2 kBest Algorithm

In HillClimber algorithm, a single best solution is saved for each service and the result of each iteration is a single list of these solutions. Since in the iterations each service is individually re-routed, the routing of each service depends on all the other existing routings in the virtual network. In kBest algorithm, however, instead of keeping a single routing for each service, $k$ solutions are saved. As shown in Figure 7.6, $k$ routings are calculated for the first service $d_1$. For each of the remaining services, $k$ routings are calculated for each $k$-best routings from the former service, yielding to $k^2$ options at each step. From

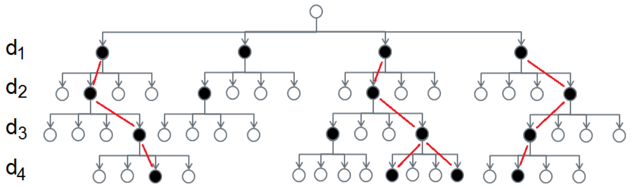


Figure 7.6: kBest algorithm example with $k = 4$

these, the best $k$ routings are selected according to the virtual network setup cost (and network utilization for PIP-Resilience). The kBest solution tree continues to branch from these nodes. At the leaves of the tree, we get $k$ best routing sequences for all the services, as shown with red lines in Figure 7.6. The lowest cost solution is then returned.

The main method of kBest algorithm with VNO-Resilience is given in Algorithm 5. The difference of PIP-Resilience is again the cost function updated with network utilization and a single virtual path calculation instead of two disjoint paths within the virtual network. As given in Algorithm 5, the kBest algorithm is also called iteratively with the difference that only the first service routing is variable since the remainder depends on this routing. If the first service routing remains unchanged or if the value of the cost function remains unchanged or if the maximum number of iterations is reached, the function returns.

**Algorithm 5 :** Shared Protection kBest Heuristic for VNO-Resilience.

```
Sort the demand pairs $d \in D$ according to their physical shortest path distance
$i = 0$ {Iteration counter}
while $i < i_{\max}$ do
  if $i = 0$ then
    $initialPathList$ = calculate_kBest_solution($D$, null)
    Calculate $cost_{\text{Virtualnetwork}}$
  else
    $d = D[0]$
    Remove the routing of $d$
    Update the costs of the links
    $pathList \leftarrow$ k_lowest_cost_disjoint_paths($d$)
    if $initialPathList == pathList$ then
      break
    end if
    calculate_kBest_solution($D$, $pathList$)
    Calculate $cost_{\text{Virtualnetwork,new}}$
    if $cost_{\text{Virtualnetwork,new}} == cost_{\text{Virtualnetwork}}$ then
      break
    else
      $cost_{\text{Virtualnetwork}} \leftarrow cost_{\text{Virtualnetwork,new}}$
    end if
  end if
end while
```

Since we need to perform $k^2$ routing calculations at each step of the kBest algorithm, the worst case complexity of this algorithm is $\mathrm{O}(i_{max}k^2|D||L||V|)$. Note that, even though the worst case complexity of the kBest algorithm is $k$ times worse than HillClimber for $k = k_w$, with $k = 3$, the difference is around 1.5-2 times since the required number of iterations are lower with kBest. This is due to the dependency of the return condition on any service routing change in case of HillClimber and only on the first service routing change in case of kBest, which has a lower probability of occurrence.

### 7.3.3 Performance Evaluation

Using the simulation setup and parameters described in Sections 7.3.1.3 and A.1.1, we evaluate in this section the performance of the proposed algorithms in comparison with the shared protection MILPs introduced in this section and dedicated protection MILPs from Chapter 5. Therefore, the same parameter settings are used as for Chapter 5, which are given in Section A.1.1. The former shows how close the performance of the heuristics is compared with the optimal solution, and the latter provides a quantitative analysis for the virtual network setup cost and network utilization gain shared protection mechanisms offer in a network virtualization environment. Finally, we discuss at which layer shared protection should be applied.

#### 7.3.3.1 Performance Evaluation of the Heuristics Compared with the Shared Protection MILP

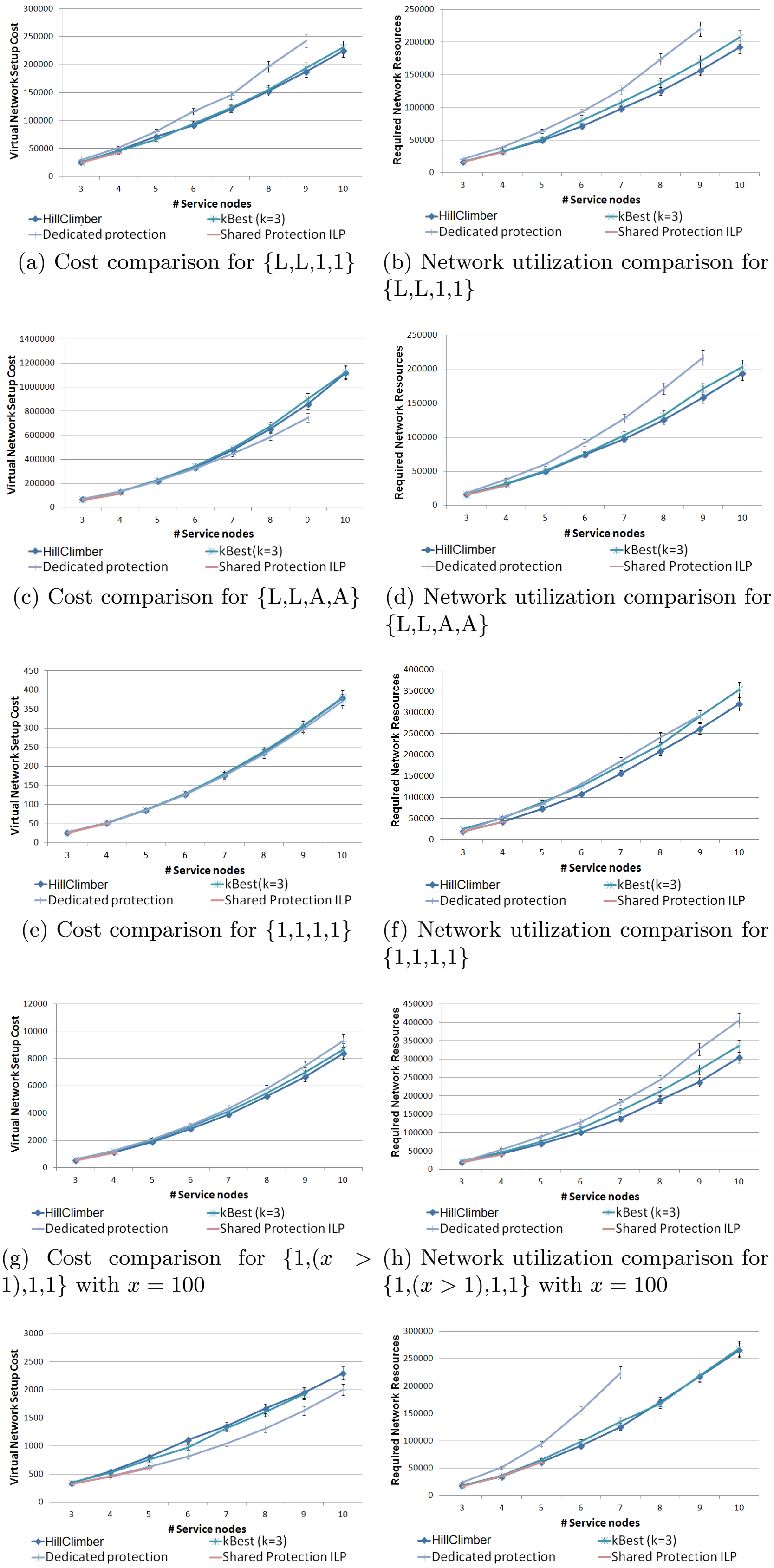
The results are organized in two sets providing the cost and network utilization values for the proposed algorithms versus the MILPs for VNO-Resilience and for PIP-Resilience. The results for VNO-Resilience and PIP-Resilience are provided in Figures 7.7 and 7.8, respectively, where the left column shows the cost results and the right column the network utilization results for each of the used cost settings. For all the cost settings, the heuristics perform very close - less than 5% difference - to the shared protection MILP except for the cost result with VNO-Resilience using the last cost setting. The reason of this problem is that having a dominant fixed cost for virtual links causes the optimal virtual network to have a minimum number of links. Using both the shared and dedicated protection MILPs generally a ring network is returned. However, since in the heuristics each service is routed consecutively, they are initially routed on the shortest connections. Therefore, the virtual network tends to have more links. The heuristics use at least one more link compared to a ring topology. All these links are utilized by more than one service, and hence, re-routing of a single service does not enable the reduction of the number of links. Finally, a higher number of links for the last cost setting causes the total cost to be higher than the MILPs. This problem does not occur with the other cost settings, where the optimal virtual networks are more densely connected.

For larger number of services, the shared protection MILP is not scalable. For already 5 service nodes and 10 services, the shared protection MILP takes hours to be solved with the solver CPLEX 12.3 [67] and the tree size of the branch and bound algorithm exceeds the memory limit on a computer with 16 cores and 60 GB RAM memory. However, both heuristic algorithms are able to solve the same problem within seconds, showing that both algorithms are both well performing and scalable. Even for virtual network designs with 10 nodes and 45 services, both algorithms require less than a minute with $k = k_w = 3$.

#### 7.3.3.2 Performance Evaluation of Shared Protection using VNO-Resilience

Figure 7.7 shows the gain of applying shared protection over dedicated protection in terms of virtual network setup cost and network utilization using VNO-Resilience. The results of the heuristics are compared with the dedicated protection MILP, and hence, the results signify at least how much gain shared protection brings for the specified settings. The gain is defined as $\frac{\text{value}_{\text{MILP}} - \text{value}_{\text{Heuristic}}}{\text{value}_{\text{Heuristic}}}$. Network utilization gain shows the reduction in the amount of the required physical resources by a virtual network designed for a given set of services. For $k = k_w = 3$, the two algorithms have results within a 5% range difference except for the last cost setting. Therefore, the results are shown with $k = 3$ for kBest but with $k_w = 50$ for HillClimber to see the effect of increased number of paths.

For cost setting {L,L,1,1}, we observe a 30% virtual network setup cost gain and 40% network utilization gain as shown in Figures 7.7a and 7.7b, respectively. Using cost setting {L,L,A,A}, the cost gain vanishes, however, a similar network utilization gain is observed as shown in Figures 7.7c and 7.7d. The reason for the cost gain to be reduced is the fact that in this cost setting the node cost has a comparable value to the link cost, which reduces



(a) Cost comparison for {L,L,1,1}

(b) Network utilization comparison for {L,L,1,1}

(c) Cost comparison for {L,L,A,A}

(d) Network utilization comparison for {L,L,A,A}

(e) Cost comparison for {1,1,1,1}

(f) Network utilization comparison for {1,1,1,1}

(g) Cost comparison for {1,($x$ > 1),1,1} with $x = 100$

(h) Network utilization comparison for {1,($x > 1$),1,1} with $x = 100$

(i) Cost comparison for {($x$ > 1),1,1,1} with $x = 100$

(j) Network utilization comparison for {($x > 1$),1,1,1} with $x = 100$

Figure 7.7: VNO-Resilience: Performance of the HillClimber ($k_w = 50$) and kBest ($k = 3$) heuristics compared with dedicated and shared protection MILPs

the relative cost reduction over the total cost heavily. In both of these cost settings, the HillClimber with $k_w = 50$ performs slightly better than the kBest and HillClimber algorithms with $k = k_w = 3$.

Using cost setting {1,1,1,1}, neither in terms of cost nor in terms of network utilization much savings are observed as shown in Figures 7.7e and 7.7f, respectively. This is due to the fact that this cost setting results in a nearly full-mesh multi-graph virtual network topology because of links and nodes having the same cost and links having fixed cost values. This reduces the opportunities for sharing the redundant resources and only 15% and 5% gains are observed in terms of network utilization using the HillClimber with $k_w = 50$ and kBest with $k = 3$, respectively.

Cost setting {1,($x > 1$),1,1} results in virtual network setup cost savings of 15% and 10% and network resource savings of 35% and 20% for HillClimber with $k_w = 50$ and kBest with $k = 3$, respectively. The emphasis on the capacity dependent link cost enables these high network resource savings, whereas the cost saving is not as high as in {L,L,1,1} due to the partial contribution of the link cost to the overall virtual network setup cost.

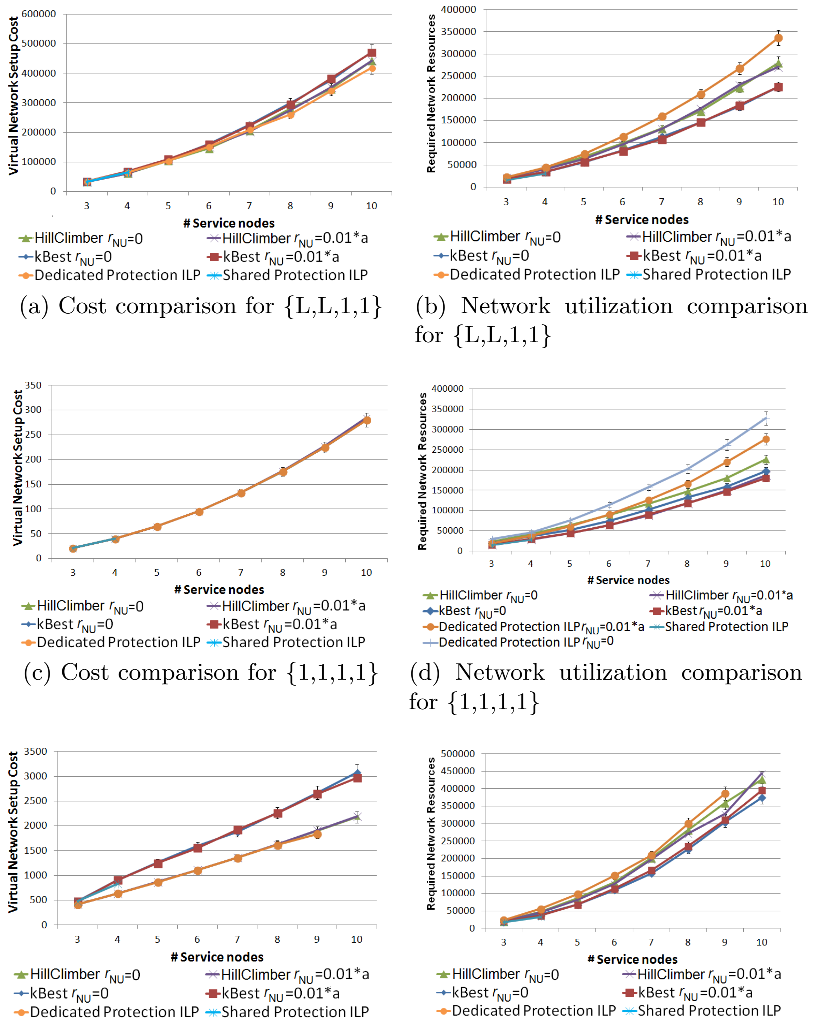
Finally, the cost setting {($x > 1$),1,1,1} results in higher virtual network setup cost using the heuristics as explained above but around 70% lower physical network resource usage due to the increased number of virtual links to enable sharing. Therefore, shared protection offers a high gain in this cost setting and the problem with the cost performance of the heuristics could be solved by enabling the re-routing of multiple services simultaneously. This would cause a trade-off between the optimality of the results and computational complexity. For this cost setting, HillClimber with $k_w = 3$ results in 10% higher cost compared to kBest. In conclusion, for VNO-Resilience both algorithms perform similar for the same $k$ values, except for the last cost setting, where HillClimber has 10% worse cost results than kBest. Increasing the number of paths from 3 to 50 yields around 10% better network utilization results for some settings with the expense of 17 times increased complexity.

Virtual network cost reduction via shared protection is observed, if the link cost depends on the physical length of the virtual link and it is higher than the node cost, or if the capacity dependent link cost is the dominant cost component as in {L,L,1,1} and {1,($x > 1$),1,1}. Still, for all the used cost settings 15-70% reduction in network resource requirement is observed, indicating the gain of applying shared protection in virtual networks.

#### 7.3.3.3 Performance Evaluation of Shared Protection using PIP-Resilience

In PIP-Resilience, the virtual network cost is expected to remain unchanged since sharing redundant resources in the physical layer does not reduce the virtual network cost. However, shared protection is expected to offer a network utilization gain, where both behaviors are seen in Figure 7.8 for the used cost settings.

Different than VNO-Resilience, in PIP-Resilience both the virtual network cost and the network utilization are minimized in the cost function to enable sharing of the redundant resources in the physical layer as described in Section 7.3.2. The emphasis of these two components are adjusted using the cost factor $r_{NU}$. Firstly, the ratio of the average virtual network cost to the network utilization is calculated for each cost setting and these values are $a = \{2.3, 3.6, 0.001, 0.04, 0.02\}$ for the used five cost settings, respectively. Therefore, using the value $a$ for a certain cost setting means having equal emphasis on both the virtual network cost and network utilization. We then vary the amount of $r_{NU}$ to see the effect of different emphasis options by using $r_{NU} = a * y$, where the $y$ value was varied as $\{0.01, 0.1, 0.5, 1, 1.5, 2\}$. For all the cost settings, the higher the $r_{NU}$ value, the more savings are observed on the network utilization side with the expense of increasing the



(a) Cost comparison for {L,L,1,1}

(b) Network utilization comparison for {L,L,1,1}

(c) Cost comparison for {1,1,1,1}

(d) Network utilization comparison for {1,1,1,1}

(e) Cost comparison for {($x > 1$),1,1,1} with $x = 100$

(f) Network utilization comparison for {($x > 1$),1,1,1} with $x = 100$

Figure 7.8: PIP-Resilience: Performance of the HillClimber ($k_w = 50$) and kBest ($k = 3$) heuristics compared with dedicated and shared protection ILPs

virtual network cost. For all the cost settings, the optimal option was therefore using $y = 0.01$, which does not cause an increase in the virtual network cost and still offers a relatively high network utilization gain. In Figure 7.8 the results of HillClimber and kBest algorithms for PIP-Resilience with $y = 0.01$ are shown.

For dedicated protection MILP, using $y > 0$ only makes a difference for the cost settings {1,1,1,1} and {1,($x > 1$),1,1}, since without network utilization minimization the virtual link mappings are chosen arbitrarily due to their fixed cost independent of their physical length. Therefore, only for these cost settings the network utilization values both with $y = 0$ and $y = 0.01$ are shown in Figure 7.8d. For the first two cost settings, since the link cost depends on the virtual link's length, cost minimization and network resource minimization are aligned, and $y = 0$ and $y = 0.01$ result in similar values. For cost setting {($x > 1$),1,1,1}, the original MILP as introduced in Chapter 5 has combinatorial complexity, and hence, a fraction of length is included to the link selection cost in our simulations. Therefore, the drawback on the network utilization due to the length-independent link cost is not observed here.

As mentioned before, shared protection in case of PIP-Resilience does not affect the virtual network cost but offers a network utilization gain. Therefore, it is of high interest for the PIPs to apply this solution for utilizing their resources more efficiently. The actual amount of the network resource savings depends on the used cost setting. In cost setting {L,L,1,1}, HillClimber algorithm offers 24% and kBest 50% network utilization gain as shown in Figure 7.8b. For {L,L,A,A}, the results are similar to Figure 7.8b with the gain being 23% and 37% for HillClimber and kBest algorithms, respectively. For cost setting {1,1,1,1}, the addition of network utilization minimization improves the network resource usage already by 20% compared with the original dedicated protection MILP as shown in Figure 7.8d. The gain of sharing is calculated by comparing the heuristic results to the dedicated protection MILP results with $y = 0.01$, and it is 50% for both algorithms. A similar result as with {1,1,1,1} is observed for the cost setting {1,($x > 1$),1,1}, where inclusion of network utilization minimization offers a gain of 22%. Additionally, shared protection reduces the network resource requirements again by 50%. For cost setting {($x > 1$),1,1,1}, the behavior of the two heuristics are different. HillClimber algorithm results in the same virtual network cost values as dedicated protection and decreases the network resource requirement slightly by around 15%. kBest algorithm, however, performs close to the shared protection MILP and offers a network utilization gain of around 30% with the trade-off of causing an increase of the virtual network cost by around 26% as shown in Figure 7.8f.

In conclusion, shared protection in case of PIP-Resilience offers high network utilization gains ranging from 15% to 50% for virtual networks with the same cost performance. For all the cost settings except {($x > 1$),1,1,1}, kBest with $k = 3$ is the preferred solution since it offers both at least as good or better results and lower computational complexity compared with HillClimber with even $k_w = 50$. In cost setting {($x > 1$),1,1,1}, however, there is a trade-off between a lower network utilization gain as offered by HillClimber algorithm and a high gain with the expense of higher virtual network cost as resulting with kBest algorithm.

#### 7.3.3.4 VNO-Resilience vs. PIP-Resilience

Finally, the answer to the question of at which layer resilience and shared protection should be applied depends on the used cost setting. For {L,L,1,1} and {($x > 1$),1,1,1}, VNO-Resilience performs better. It results in 90% lower virtual network cost and 35% lower network resource requirement with {L,L,1,1}, and 50% lower network resource requirement with the same virtual network cost with {($x > 1$),1,1,1}, compared with PIP-Resilience. For cost setting {1,1,1,1}, it is better to apply PIP-Resilience, which results in 15% lower cost and 30% lower network resource requirement compared with VNO-Resilience. For {L,L,A,A} and {1,($x > 1$),1,1}, there is a trade-off between cost and network utilization performance. With the former, VNO-Resilience results in 30% higher virtual network cost but 20% lower network resource requirement and with the latter 10% reduced virtual network cost but 40% higher network resource requirement.

## 7.4 Shared Protection in Virtual Networks with Combined Optimization

In this section, the application of shared protection on virtual network architectures for cloud services is investigated. Similar to the case with connectivity services, the redundant virtual link resources are shared between different services or virtual networks. In addition, sharing of redundant virtual IT resources is also enabled. This can be realized by keeping a backup of the primary site VMs and re-instantiating these VMs on the shared idle resources of the DR site in case of failure or even before a natural disaster like a hurricane

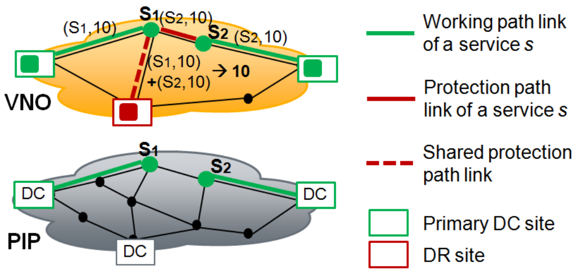

(S1,10)
S1(S2,10)
S2 (S2,10)
(S1,10)
+(S2,10)
→ 10
VNO
S1
S2
DC
DC
DC
PIP
Working path link of a service s
Protection path link of a service s
Shared protection path link
Primary DC site
DR site

Figure 7.9: Example for sharing of redundant virtual network and DC resources: Due to the physical disjointness of the working paths and primary DCs of the two services, the protection path and DC resources can be shared for them.

hits the area to avoid service interruption. In the following subsections, first, optimization models allowing resilient virtual network design with shared protection for cloud services are introduced. Then, we present the heuristic models and evaluate their performance. Note that throughout this section the terms anycast service and cloud service are used interchangeably.

### 7.4.1 Optimization Models with Shared Protection for Cloud Services

This subsection introduces virtual network design models with shared protection, where both IT and network resources are optimized simultaneously. The problem definition is given in the following. (i) A virtual network topology $G_l(V, L)$ with all the virtual link and node candidates, (ii) a physical topology with DC locations and (iii) the set of anycast (cloud) service requests with their bandwidth and virtual machine requirements are given. The aim is to find a minimum cost resilient virtual network topology with attached DCs having protection against single link and DC failures such that (i) all the service requests are satisfied and (ii) sharing of redundant network and DC resources is maximized.

Shared protection in virtual networks reduces possibly both the virtual network cost and required network and DC capacities by sharing the protection resources between the services. The main requirement for shared protection is that the working capacities of two services, namely both the primary DC sites and the working paths connecting each service node to its primary DC site, should be mapped on physically disjoint resources. In that case, the common protection link resources and DR site resources can be shared as shown in Figure 7.9, where the green links and DC sites show the working paths and primary DCs and the red ones are the protection resources. In the case of this example, the bandwidth on the dashed link and the VM in the DR site can be shared by these two services. If the disjointness criteria are not met, the primary resources might fail simultaneously making sharing impossible.

In virtual networks, resilience provisioning and redundant resource sharing can be done either in the virtual or in the physical layers. The details of these two models and a list of the sets, parameters and variables used in both models and in the heuristics are given in the following.

- Sets:
  - $S$: Set of the service nodes
  - $C$: Set of the DC connection nodes
  - $V$: Set of the all virtual nodes with $S \cup C = V$ and $S \cap C = \{\}$
  - $L$: Set of the virtual link candidates, where there is at least one link between all node pairs in $S$ and from each node in $S$ to all nodes in $C$
  - $D$: Set of all possible realizations of the requested anycast services, where $D = S \times C$

- $D_s$: Set of the requested anycast services from a service node $s \in S$ with $|D_s| = C$ and $D_s \subseteq D$
- $E_l$: Set of the endpoints of link $l \in L$
- $Z$: Set of virtual link pairs $(l, k) \in L^2$, which share at least one physical edge, i.e., which are not link disjoint
- $E$: Set of the physical links in the physical network topology
- $P_l$: Set of the physical links $e \in E$, on which the virtual link $l \in L$ is mapped
- $R$: Set of the DC sites which are in the same region
- $\Sigma$: Set of the regions $\Xi \in \Sigma$, where each region $\Xi$ is again a set of the DCs which are in that region
- $W_l$: Set of the physical links $e \in E$, on which the primary path of virtual link $l \in L$ is mapped
- $B_l$: Set of the physical links $e \in E$, on which a backup path of the virtual link $l \in L$ is mapped
- $\Lambda$: Set of DCs with $\Lambda_p \in \Lambda$ and $\Lambda_b \in \Lambda$ denote a primary or DR site if selected as such, respectively

- Parameters:
  - $n_{\text{dc}}$: Number of the DCs to be used in total for a service with $n_{\text{dc}} \in \{2, ..., |C|\}$
  - $b_d$: Requested bandwidth for the service $d \in D$
  - $n_d$: Requested network node resources for the service $d \in D$
  - $r_d$: Requested server resources for the service $d \in D$
  - $t_l$: Physical length of link $l \in L$
  - $\lambda_l$: Fixed setup cost for having a new link $l \in L$
  - $\theta_l$: Setup cost per unit capacity for link $l \in L$
  - $\mu_v$: Fixed setup cost for having a network node $v \in V$
  - $\eta_v$: Setup cost per unit capacity for network node $v \in V$
  - $\phi_c$: Fixed setup cost for having a new server connected to node $c \in C$
  - $\varphi_c$: Setup cost per unit capacity for a server connected to node $c \in C$
- Variables:
  - $a_{s,c}$: Binary variable taking the value of 1 if the node pair (s,c) is used to satisfy the anycast demand with source $s \in S$ and the DC connected to $c$ is the primary site, 0 otherwise
  - $a'_{s,c}$: Binary variable taking the value of 1 if the node pair (s,c) is used to satisfy the anycast demand with source $s \in S$ and the DC connected to $c$ is a DR site, 0 otherwise
  - $\beta_{i,d,l}$: Binary variable taking the value of 1 if the link $l \in L$ is used for the demand $d \in D$ and if demand $d = (s, c)$ is chosen as one of the realizations of the anycast service from $s$, 0 otherwise. The index $i$ is used to distinguish the working and protection paths, $i = 1$ is the working path and $i > 1$ are the protection paths.
  - $\delta_{d,v}$: Binary variable taking the value of 1 if the node $v \in V$ is used for the demand $d \in D$ and if demand $d = (s, c)$ is chosen as one of the realizations of the anycast service from $s$, 0 otherwise
  - $\gamma_l$: Binary variable taking the value of 1 if the link $l \in L$ is in the resulting virtual network, 0 otherwise
  - $\alpha_v$: Binary variable taking the value of 1 if the node $v \in V$ is in the resulting virtual network, 0 otherwise
  - $y_c$: Binary variable taking the value of 1 if the DC connected to node $c \in C$ is in the resulting virtual network, 0 otherwise
  - $u_l \in [0, \infty]$: Used capacity on link $l \in L$
  - $\omega_v \in [0, \infty]$: Used capacity on node $v \in V$
  - $z_c \in [0, \infty]$: Used capacity on DC connected to node $c \in C$

- $\phi_{l,l'} \in [0, \infty]$: Used capacity on link $l' \in L$ that is used for the protection of the link $l \in L$
- $\phi_l \in [0, \infty]$: Used protection capacity on link $l \in L$
- $\psi_l \in [0, \infty]$: Used working capacity on link $l \in L$
- $\tau_{d_1,d_2,l,l'}$: Binary variable taking the value of 1 if the link $l' \in L$ is used for the protection of the link $l \in L$, where $l'$ is part of the path of $d_2 \in D_s$ and it is a protection path, i.e. $i \geq 2$, and $l$ is part of the path of $d_1 \in D_s$ and it is the primary path, 0 otherwise
- $\tau_{d,c,l}$: Binary variable taking the value of 1 if the link $l \in L$ is part of the protection path of $d \in D_s$ and the server $c \in C$ is used as the primary site of $d$, 0 otherwise
- $\phi_{c,l} \in [0, \infty]$: Used capacity on link $l \in L$ that is used for the protection of the server $c \in C$
- $n_{c,c'} \in [0, \infty]$: Used capacity on server $c' \in C$ that is used for the protection of the server $c \in C$
- $n_c \in [0, \infty]$: Used protection capacity on server $c \in C$
- $v_c \in [0, \infty]$: Used working capacity on server $c \in C$
- $g_{s,c,c'}$: Binary variable taking the value of 1 if the server $c' \in C$ is used for the protection of the server $c \in C$ for the anycast service with source node $s \in S$, 0 otherwise
- $\xi_{d,l,c}$: Binary variable taking the value 1 if a service $d = (s, c) \in D_s$ used the DC $c$ as its backup DC and the virtual link $l \in L$ is used as part of the service's working path, 0 otherwise
- $\kappa_{c,l}$: Used capacity on server $c \in C$ that is used for the protection of the servers with working paths containing $l \in L$
- $\kappa_c$: Used capacity on server $c \in C$ that is used for the protection of the servers calculated according to the working path disjointness criterion
- $\varepsilon$: Setup cost of the virtual network
- $\epsilon_{1,d,e}$: Binary variable taking the value 1 if a physical edge $e \in E$ is used in the working path mapping of the route of a service $d \in D$, 0 otherwise
- $\epsilon_{2,d,e}$: Binary variable taking the value 1 if a physical edge $e \in E$ is used in the backup path mapping of the route of a service $d \in D$, 0 otherwise

#### 7.4.1.1 VNO-Resilience

In VNO-Resilience, as described in Chapter 6, the services are routed in the virtual layer to $n_{\text{dc}}$ different server locations. An example with $n_{\text{dc}} = 2$ locations is shown in Figure 6.6a. Both the servers and the paths leading to these servers have to be physically disjoint, s.t. in case of a failure at the primary site, the DR site can be used by re-routing the service there in the virtual network. The sharing of the protection resources is realized in the virtual layer as well by reusing these redundant resources for different services if the disjointness criteria of the primary resources are met.

The constraints for the VNO-Resilience model are given in the following. To enable sharing of the protection resources, there is the need to differentiate among the primary and protection resources for each anycast demand within the MILP. Equation (7.26) ensures that the anycast server with source $s$ is routed to one primary server site. Equation (7.27) similarly ensures the existence of $n_{\text{dc}} - 1 \in \{2, ..., |C|\}$ paths leading to DR site locations for the anycast service with source $s$.

$$\sum_{c \in C} a_{s,c} = 1 \quad \forall s \in S \tag{7.26}$$

$$\sum_{c \in C} a'_{s,c} = n_{\text{dc}} - 1 \quad \forall s \in S \tag{7.27}$$

Equations (7.28) and (7.29) are the link-flow constraints for the primary and protection paths of the anycast services.

$$\sum_{l\in L:v\in E_l} \beta_{1,d,l} = \begin{cases} a_{s,c} & \text{if } v = s \text{ or } v = c \\ 2\delta_{d,v} & \text{otherwise} \end{cases} \qquad \forall d = (s,c) \in D,\ v \in V \tag{7.28}$$

$$\sum_{l\in L:v\in E_l} \beta_{i,d,l} = \begin{cases} a'_{s,c} & \text{if } v = s \text{ or } v = c \\ 2\delta_{d,v} & \text{otherwise} \end{cases} \qquad \forall d = (s,c) \in D,\ v \in V,\ i \in \{2,..,k\} \tag{7.29}$$

Since the assignment of primary and protection paths is done within the MILP, (7.30) is introduced to ensure that the primary path's propagation delay is limited by the average delay of the protection paths. If 1:1 protection is used, it guarantees that the primary path has equal or lower delay than the protection path.

$$\sum_{l\in L} \beta_{1,d_1,l}\, t_l \le \frac{1}{n_{\mathrm{dc}}-1} \sum_{i\in\{2,..,n_{\mathrm{dc}}\}} \sum_{l\in L} \sum_{d_2\in D_s:d_1\neq d_2} \beta_{i,d_2,l}\, t_l \qquad \forall s \in S,\ d_1 \in D_s \tag{7.30}$$

For the link, node and virtual machine usage indicators and for node capacity calculation the constraints (6.10)-(6.13) and (6.18) from Chapter 6 are used with the difference that the sum $a_{s,c} + a'_{s,c}$ is inserted instead of $a_{s,c}$. Diversity constraints (6.20) and (6.21) from Chapter 6 are also directly adopted for link and node diversity. DC diversity constraints need to be updated due to the differentiation of primary and backup sites and are given in (7.31) and (7.32). The diversity constraints can be easily extended for multiple and regional failures by generating the set $Z$ accordingly.

$$a_{s,c_1} + a'_{s,c_2} \le 1 \quad \forall s \in S,\ (c_1,c_2) \in R \tag{7.31}$$

$$a'_{s,c_1} + a'_{s,c_2} \le 1 \quad \forall s \in S,\ (c_1,c_2) \in R : \{c_1 \neq c_2\} \tag{7.32}$$

The resources of the protection paths can be shared, if the primary paths using these protection paths are mutually disjoint and if the DR sites are disjoint. Firstly, the constraints due to the link disjointness requirement will be given. The constraints (7.33) and (7.34) are used to determine which links are used to protect which ones in the routing of which service. As given in (7.35), the required capacity on a link $l'$ for protection of another link $l$ is calculated as the sum of the bandwidth requests of all anycast services having link $l$ in their primary and link $l'$ in one of their protection paths.

$$\beta_{1,d_1,l} + \beta_{i,d_2,l'} \le 1 + \tau_{d_1,d_2,l,l'} \quad \forall s \in S,\ (d_1,d_2) \in D_s^2,\ d_1 \neq d_2,\ l,l' \in L : l \neq l',\ i \in \{2,..,n_{\mathrm{dc}}\} \tag{7.33}$$

$$2\,\tau_{d_1,d_2,l,l'} \le \beta_{1,d_1,l} + \beta_{i,d_2,l'} \quad \forall s \in S,\ (d_1,d_2) \in D_s^2,\ d_1 \neq d_2,\ l,l' \in L : l \neq l',\ i \in \{2,..,n_{\mathrm{dc}}\} \tag{7.34}$$

$$\phi_{l,l'} = \sum_{(d_1,d_2)\in D_s^2:d_1\neq d_2} b_d\, \tau_{d_1,d_2,l,l'} \quad \forall s \in S,\ l,l' \in L^2 : l \neq l' \tag{7.35}$$

The required protection capacity on a link $l'$ is at least equal to or greater than the maximum of the required protection capacities due to the links using $l'$ as a protection link if these links are mutually disjoint as given in (7.36). In case of non-disjointness, the maximum of the sums of the link capacities, which share a physical link, are considered as given in (7.37).

$$\phi_{l'} \geq \phi_{l,l'} \quad \forall l, l' \in L : l \neq l' \tag{7.36}$$

$$\phi_{l'} \geq \sum_{l:e \in P_l} \phi_{l,l'} \quad \forall l' \in L,\ e \in E : e \notin P_{l'} \tag{7.37}$$

The capacity requirement of the protection links due to the primary DC disjointness needs to be considered as well since a failure of a DC will cause all the services using this DC to be rerouted to their DR sites using their protection paths. In that case, the protection capacity on those links cannot be shared. To enable this calculation, we need to relate the primary DCs to the protection link usage as given in (7.38)-(7.42).

$$\beta_{2,d,l} + a_{s,c} \leq 1 + \tau_{d,c,l} \quad \forall s \in S,\ d \in D_s,\ c \in C,\ l \in L \tag{7.38}$$

$$2\,\tau_{d,c,l} \leq a_{s,c} + \beta_{2,d,l} \quad \forall s \in S,\ d \in D_s,\ c \in C,\ l \in L \tag{7.39}$$

$$\phi_{c,l} = \sum_{d \in D} b_d\, \tau_{d,c,l} \quad \forall c \in C,\ l \in L \tag{7.40}$$

$$\phi_l \geq \phi_{c,l} \quad \forall c \in C,\ l \in L \tag{7.41}$$

$$\phi_l \geq \sum_{c:c \in \Xi} \phi_{c,l} \quad \forall l \in L,\ \Xi \subset \Sigma \tag{7.42}$$

Equation (7.43) is the constraint used to calculate the capacity requirements due to the primary paths on each link. Finally, the total capacity required on a link is calculated as the sum of the working and protection capacity on that link as given in (7.44).

$$\psi_l = \sum_{d \in D} \beta_{1,d,l}\, b_d \quad \forall l \in L \tag{7.43}$$

$$u_l = \phi_l + \psi_l \quad \forall l \in L \tag{7.44}$$

Redundant DC capacity sharing is enabled by the following constraints. Constraints (7.45) and (7.46) are used to determine if a server is used as the DR site of another one for service $s$.

$$a_{s,c} + a'_{s,c'} \leq 1 + g_{s,c,c'} \quad \forall s \in S,\ c, c' \in C^2 : c \neq c' \tag{7.45}$$

$$2\,g_{s,c,c'} \leq a_{s,c} + a'_{s,c'} \quad \forall s \in S,\ c, c' \in C^2 : c \neq c' \tag{7.46}$$

Equation (7.47) is used to calculate the protection capacity needed on a DC for each DC using it as a DR site for all the services. (7.48) and (7.49) calculate the protection capacity required on a server as the maximum capacity of the servers using it if they are physically disjoint and the sum of the capacities for the servers, which are in the same region.

$$n_{c,c'} = \sum_{s \in S\,:\, d=(s,c)} r_d\, g_{s,c,c'} \quad \forall c, c' \in C^2 : c \neq c' \tag{7.47}$$

$$n_{c'} \geq n_{c,c'} \quad \forall c, c' \in C^2 : c \neq c' \tag{7.48}$$

$$n_{c'} \geq \sum_{c \in \Xi} n_{c,c'} \quad \forall c' \in C, \ \Xi \subset \Sigma \tag{7.49}$$

Additionally, the disjointness of the working paths of the services using the same DR site need to be also taken into account since non-disjoint working paths could cause the services to loose access to their primary DCs simultaneously. For this purpose, first we need to relate the working paths of the services to their usage of a DC $c$ as their DR site, which is given in (7.50) and (7.51).

$$\beta_{1,d,l} + a'_{s,c} \leq 1 + \xi_{d,l,c} \quad \forall s \in S, \ d \in D_s, \ c' \in C, \ l \in L \tag{7.50}$$

$$2\,\xi_{d,l,c} \leq a'_{s,c} + \beta_{1,d,l} \quad \forall s \in S, \ d \in D_s, \ c' \in C, \ l \in L \tag{7.51}$$

In (7.52) the sum of the required capacities on a DC $c$ due to different services using the working link $l$ are calculated. Finally, (7.53) and (7.54) calculate the required protection capacity on DC $c$ due to the shared working paths of the different services analogous to (7.41) and (7.42). The maximum of the capacity requirements due to the DC and path disjointness of the services determines the protection capacity value on DC $c$ as given in (7.55). Finally, (7.56) is used to calculate the working capacity on each DC and the total required capacity on each DC is equal to the sum of the working and protection capacity on that DC as given in (7.57).

$$\kappa_{c,l} = \sum_{d \in D} r_d \, \xi_{d,l,c} \quad \forall c \in C, \ l \in L \tag{7.52}$$

$$\kappa_c \geq \kappa_{c,l} \quad \forall c \in C, \ l \in L \tag{7.53}$$

$$\kappa_c \geq \sum_{l : e \in P_l} \kappa_{c,l} \quad \forall c \in C, \ e \in E \tag{7.54}$$

$$n_c \geq \kappa_c \quad \forall c \in C \tag{7.55}$$

$$v_c = \sum_{s \in S : d = (s,c)} r_d \, a_{s,c} \quad \forall c \in C \tag{7.56}$$

$$z_c = n_c + v_c \quad \forall c \in C \tag{7.57}$$

The objective function used in VNO-Resilience minimizes the virtual network setup cost as given in (7.61). As introduced in Chapter 6, the cost of the virtual network constitutes of the link cost, network node cost and VM cost, where each of them has again two parts, namely the fixed setup cost for having a new link, node or VM in the virtual network and the capacity dependent cost depending on the requested capacity on that link, node or VM. Minimizing the virtual network cost implicitly allows to optimize for shared protection because the redundant link and VM capacities are shared between the services in the virtual layer lowering the virtual network cost.

$$\varepsilon_l = \lambda_l \gamma_l + \theta_l u_l \quad \forall l \in L \tag{7.58}$$

$$\varepsilon_v = \mu_v \alpha_v + \eta_v \omega_v \quad \forall v \in V \tag{7.59}$$

$$\varepsilon_c = \phi_c y_c + \varphi_c z_c \quad \forall c \in C \tag{7.60}$$

$$\min\ \varepsilon,\ \varepsilon = \sum_{l \in L} \varepsilon_l + \sum_{v \in V} \varepsilon_v + \sum_{c \in C} \varepsilon_c \tag{7.61}$$

#### 7.4.1.2 PIP-Resilience

In PIP-Resilience, providing resilience is the responsibility of the PIP(s). As described in Chapter 6, the services are routed on a single path in the virtual network layer to the primary DC site. This virtual path is protected by the PIP, where each virtual link has a 1:1 protection mapping on the physical layer. Moreover, the dcPIP owning the primary site is responsible for providing the DC resilience. For this purpose the DR site(s) for each DC candidate and their resilient physical connection path(s) are pre-calculated. This information is incorporated in the fixed cost factor of the primary VMs. The VNO sees only a single virtual path connected to a single DC site, which are protected in the physical layer. The re-routing to the DR site in case of a failure is realized in the physical layer by the corresponding PIP as shown in Figure 6.6b. As a result, the virtual topology remains unchanged and ideally the services are not disrupted.

In PIP-Resilience protection capacity of the resilient virtual link mappings is shared between different services. Additionally, the protection resources on the DR sites are also shared if the disjointness criteria are met. The main difference of the PIP-Resilience model compared with VNO-Resilience is that in PIP-Resilience only a single DC site is chosen as the primary site for each service, and hence, the DR sites are not visible to the VNO. The selection of a single primary site is ensured with (7.26), and (7.27) is omitted. Since the resilience is provided in the physical domain, the diversity constraints are excluded from the model. The constraints (6.9)-(6.19) from Chapter 6 are directly adopted, which deal with routing of the services and selection of the necessary virtual components.

In PIP-Resilience, redundant resource sharing is done in the physical layer. Therefore, for link capacity sharing the virtual routes need to be mapped onto the physical routes, which is realized using (7.62) and (7.63). Constraints (7.64) and (7.65) are used to ensure that the $\epsilon$ values are set to 0 if no virtual link mapped on them is used. The remainder of the redundant link capacity calculation is analogous to VNO-Resilience, where $y_e$ is the sum of the working and protection capacity on each edge $e$.

$$\epsilon_{1,d,e} = \beta_{1,d,l} \quad \forall l \in L,\ e \in W_l,\ d \in D \tag{7.62}$$

$$\epsilon_{2,d,e} = \beta_{1,d,l} \quad \forall l \in L,\ e \in B_l,\ d \in D \tag{7.63}$$

$$\epsilon_{1,d,e} \leq \sum_{l \in L : e \in W_l} \beta_{1,d,l} \quad \forall d \in D,\ l \in L,\ e \in W_l \tag{7.64}$$

$$\epsilon_{2,d,e} \leq \sum_{l \in L : e \in B_l} \beta_{1,d,l} \quad \forall d \in D,\ l \in L,\ e \in B_l \tag{7.65}$$

Similar to link capacity sharing, DC capacity sharing constraints are analogous to VNO-Resilience model with the difference that in PIP-Resilience for each primary DC candidate a DR site is pre-calculated and assigned. Therefore, we use the constraint (7.66), which sets the value of the variable $g_{s,c,c'}$ for a DC $c'$ and service from the source node $s$ as 1 if the DC $c$ is the primary DC of the service $s$, and $c'$ is assigned as one of the DR sites for DC $c$.

$$g_{s,c,c'} = a_{s,c} g_{c,c'} \quad \forall s \in S,\ c, c' \in C^2 : c \neq c' \tag{7.66}$$

To enable the optimization of the network and DC resource utilization in the physical layer, the total capacity required on the physical edges and the capacity required on the DCs are added to the minimization function with their corresponding weighting coefficients $r_{NU}$ and $r_{DC}$ for network utilization and DC utilization, respectively, as given in (7.67).

$$\min \left( \varepsilon + r_{NU} \sum_{e \in E} y_e + r_{DC} \sum_{c \in C} z_c \right) \tag{7.67}$$

### 7.4.2 Heuristics for Shared Protection in Virtual Networks for Cloud Services

The general logic behind the proposed heuristics realizing virtual network design for cloud services is based on the existing heuristics for shared protection in the physical layer [137, 138] and anycast routing heuristics [61]. In the existing heuristics, first a subset of DC sites are selected for routing and then the services are routed iteratively. The routing process is analogous to the unicast case, where the cost adjustment of a link, node or VM depends on its current usage during a new service routing. Moreover, for shared protection mechanism to be added to the virtual network design, the cost calculation needs to be adapted accordingly for both virtual links and VMs. In case of redundant link resources sharing, if a link is already used and is used for the protection path of a service $d_1 \in D$ and the working path of a new service $d_2$ is disjoint with the working path of $d_1$, the cost of this link is reduced to zero. Similarly the VM cost is also adapted to enable sharing of redundant DC resources. The node capacity is not shared since we only have protection against single physical (and implicitly virtual) link and DC failures.

#### 7.4.2.1 HillClimber Algorithm

In the HillClimber algorithm, a Greedy approach is used. The services are routed iteratively and at each service routing the new DC location and path pairs for primary and DR sites, which offer a minimum virtual network cost, are selected. Once all the services are routed, the algorithm iterates by re-routing each single service keeping the other service routings and checks if the new solution offers a better cost. If in one iteration all the service routings remain unchanged or if the maximum number of the iterations, $i_{max}$, are reached the algorithm returns. Algorithm 6 shows the version for VNO-Resilience. In case of PIP-Resilience, the only difference is that instead of the virtual network cost, the summation of the cost with network utilization and DC utilization is minimized as given in (7.67).

In both HillClimber and kBest algorithms the same method is used for service routings, which are shown in Algorithm 7 and 8. Firstly, $n$ minimum cost primary DC candidates are selected according to the total cost of the DC and its connection path. Then, for each of these primary DCs, a DR site and its connection path is selected using Algorithm 8 from $x$ DR site candidates. To enable sharing, the costs of the virtual network and IT resources are updated according to their current usage. If they are used for protection purposes and if they can be shared for the current service protection, the cost is set to 0. The sharing condition for a protection link is the disjointness of the primary paths as well as the primary DCs of the services using this protection link. Similarly, for sharing of the DR site resources, disjointness of the primary DCs and the primary paths are checked.

**Algorithm 6** Anycast Shared Protection HillClimber Heuristic for VNO-Resilience

```
for $i$ from 0 to $i_{max}$ do
    if $i == 0$ then
        for all $s \in S$ do
            DClist $\leftarrow \mathbf{0}$, NodeList $\leftarrow \mathbf{0}$
            $(p_{\text{protection}}, p_{\text{working}}) \leftarrow$ lowest_cost_disjoint_paths(VNet, $s$, DClist, NodeList)
            set_properties ([DClist], [$p_{\text{protection}}, p_{\text{working}}$], [NodeList])
        end for
        Calculate $\varepsilon$ (% Virtual network setup cost)
    else
        $changed \leftarrow$ false
        for all $s \in S$ do
            Clear current routing of $s$
            DClist $\leftarrow \mathbf{0}$, NodeList $\leftarrow \mathbf{0}$
            $(p_{\text{protection}}, p_{\text{working}}) \leftarrow$ lowest_cost_disjoint_paths(VNet, s, DClist, NodeList)
            calculate $\varepsilon_{\text{new}}$
            if $\varepsilon_{\text{new}} < \varepsilon$ then
                $\varepsilon \leftarrow \varepsilon_{\text{new}}$
                $changed \leftarrow$ true
                set_properties ([DClist], [$p_{\text{protection}}, p_{\text{working}}$], [NodeList])
            else
                Reroute $s$ according its previous routing
            end if
        end for
        if $changed = 0$ then
            break
        end if
    end if
end for
```

Note that for PIP-Resilience only the primary DC and its connecting virtual links, with their primary and protection path mappings are selected with a similar logic.

The lowest cost working and protection path calculation algorithms used in Algorithm 7 and 8 are based on the Bellmann-Ford algorithm. Connection paths are calculated for each of the working DC candidates ($|C|$ path calculations) to determine the best $n$ primary DCs and connection paths and $n\,x$ protection path calculations are done in total for $x$ DR sites and for each of the $n$ primary DCs. This procedure is repeated in each iteration for all the services in $S$, yielding a total worst-case complexity in the order of $\mathrm{O}(i_{max}|S|(|C| + n\,x)|L||V|)$.

#### 7.4.2.2 kBest Algorithm

In HillClimber algorithm, a single best solution is saved for each service and the result of each iteration is a single list of these solutions. Since in the iterations each service is individually re-routed, the routing of each service depends on all the other existing routings in the virtual network. In kBest algorithm, as described for the unicast case, instead of keeping a single routing for each service, $k$ solutions are saved. As shown in Figure 7.6, $k$ routings are calculated for the first service $d_1$. For each of the remaining services, $k$ routings are calculated for each $k$-best routings from the former service, yielding to $k^2$ options at each step. From these, the best $k$ routings are selected according to the virtual network setup cost (and network and DC utilization for PIP-Resilience). The kBest solution tree continues to branch from these nodes. At the leaves of the tree, we get $k$ best routing sequences for all the services, as shown with red lines in Figure 7.6. The lowest cost solution is then returned. In the iterations of the algorithm, the first service routing is re-done keeping the other service routings. If the first routing changes and yields a lower cost, the other services are also re-routed. The algorithm iterates until the first service routing does not change or yields a higher cost or the maximum number of iterations is reached. Therefore, the worst-case complexity of the kBest algorithm is $k$ times higher than the HillClimber algorithm and is $\mathrm{O}(i_{max}k|S|(|C| + n\,x)|L||V|)$.

**Algorithm 7** lowest_cost_disjoint_paths(Virtual network, $s$, DClist, NodeList)

```
for all $\Lambda_p \in \Lambda$ do
    $P_{index} \leftarrow$ Add index of $\Lambda_p$, $i_{\Lambda_p}$
    $(P_{path}, P_{node}, P_{cost}) \leftarrow$ Add min_cost_working_path$(s, \Lambda_p)$
end for
$Sorted\_P_{index} \leftarrow$ Sort$(P_{index})$ in ascending order
Populate $B_{index}$ with the first $x$ DCs' indices from $Sorted\_P_{index}$ and with DCs already used as backup DC
for all t $\leq$ n do
    $i_{\Lambda_p} \leftarrow Sorted\_P_{index}[t]$
    Update the costs according to $P_{path}[i_{\Lambda_p}]$
    for all j $\leq$ $B_{index}$.length do
        $i_{\Lambda_b} \leftarrow B_{index}[j]$
        if $\Lambda_b$ disjoint area from $\Lambda_p$ then
            $T_{emp}B_{index} \leftarrow i_{\Lambda_b}$
            $(T_{emp}B_{path})\leftarrow$ min_cost_disjoint_protection_path $(s, \Lambda_p, \Lambda_b, P_{path}[i_{\Lambda_p}], P_{node}[i_{\Lambda_p}])$
        end if
    end for
    $B_{final} \leftarrow$ Add minimum cost path and DC pair index $i_{min,\Lambda_b}$
    $(B_{path}$,$B_{node}$,$B_{cost}) \leftarrow$ add $(T_{emp}B_{path}[i_{min,\Lambda_b}]$, $T_{emp}B_{node}[i_{min,\Lambda_b}]$, $T_{emp}B_{cost}[i_{min,\Lambda_b}])$
    Clear $T_{emp}B$ lists
    Clear the routed working path $P_{path}[i_{\Lambda_p}], P_{node}[i_{\Lambda_p}]$ from virtual network
end for
$(p_{\text{protection}}, p_{\text{working}}) \leftarrow$ min cost $(B_{path}$, $P_{path})$
(DClist) $\leftarrow$ min cost($\Lambda_b$ of $B_{final}$, $\Lambda_p$ of $P_{final}$)
(NodeList) $\leftarrow$ min cost $(B_{node}$, $P_{node})$
return $(p_{\text{protection}}, p_{\text{working}})$
```

### 7.4.3 Performance Evaluation

In this subsection, the performance evaluation of the shared protection models for cloud services is presented, which has been done using extensive simulations with the same settings for Chapter 6 as shown in Section A.1.4. The aim of the simulations is threefold. The first aim is to compare the results of the heuristics with the dedicated and shared protection MILPs to evaluate their performance. The second aim is to answer the question of how much gain shared protection brings in a network virtualization architecture for cloud services. And finally, the third aim is to evaluate at which layer shared protection should be applied.

#### 7.4.3.1 Performance of the Heuristics

The proposed heuristics are scalable and perform close to optimal. For small virtual network instances, where the shared protection MILP can still run, the difference of the heuristics to the MILP is less than 5%. For larger instances, a comparison with dedicated protection is provided. In case of PIP-Resilience, the virtual network setup cost is not affected by shared protection and ideally should remain unchanged. Very low variation of the virtual network cost values with the heuristics show that they also perform well for larger instances. Finally, while shared protection MILP lasts hours for more than 3 DCs with solver CPLEX 12.3 and the solution tree exceeds the memory limits of a computer with 16 cores and 60GB RAM memory; even for virtual network designs with 50 service source nodes and 6 DCs, both heuristics can compute the solution within 2 minutes on a random physical topology with 100 nodes. Due to the higher number of iterations needed by HillClimber, the average performance of the two algorithms is comparable. For the simulations the $k$ value in kBest algorithm is taken as 3.

#### 7.4.3.2 Virtual Network Setup Cost and Resource Utilization Gain of Shared Protection over Dedicated Protection

The amount of redundant resource saving depends on the used cost setting and resilience model. Using the proposed heuristics, PIP-Resilience allows more sharing than VNO-

**Algorithm 8** min_cost_disjoint_protection_path $(s, \Lambda_p, \Lambda_b, P_{path}[i_{\Lambda_p}], P_{node}[i_{\Lambda_p}])$

```
for $l \in L$ do
    if $w_l = 0$ and $p_l = 0$ then
        $cost_l \leftarrow \lambda_l + \theta_l b_d$
    else if $w_l = 1$ and $p_l = 0$ then
        $cost_l \leftarrow \theta_l b_d$
    else
        $b_{\mathrm{l_E,protection,new}} \leftarrow b_{\mathrm{l_E,protection}}$
        for $e \in B'_l : l' \in P_{path}[i_{\Lambda_p}]$ do
            if $M[i_l][i_e] + b_d > b_{\mathrm{l,protection,new}}$ then
                $b_{\mathrm{l_E,protection,new}} \leftarrow M[i_l][i_e] + b_d$
            end if
        end for
        $b_{\mathrm{l_\Lambda,protection,new}} \leftarrow b_{\mathrm{l_\Lambda,protection}}$
        if $N[i_l][i_{\Xi_{\Lambda_p}}] + b_d > b_{\mathrm{l_\Lambda,protection,new}}$ then
            $b_{\mathrm{l_\Lambda,protection,new}} \leftarrow N[i_l][i_{\Xi_{\Lambda_p}}] + b_d$
        end if
        $b_{\mathrm{l,\,protection,new}} \leftarrow \max(b_{\mathrm{l_E,protection,new}}, b_{\mathrm{l_\Lambda,protection,new}})$
        $cost_l \leftarrow \theta_l(b_{\mathrm{l,protection,new}} - b_{\mathrm{l,protection}})$
    end if
end for
for $v \in V$ do
    if $v_{flag} = 0$ then
        $cost_v \leftarrow \mu_v + \eta_n n_d$
    else
        $cost_v \leftarrow \eta_n n_d$
    end if
end for
if $w_{\Lambda_b} = 0$ and $p_{\Lambda_b} = 0$ then
    $cost_{\Lambda_b} \leftarrow \phi_{\Lambda_b} + \varphi_{\Lambda_b} r_d$
else if $w_{\Lambda_b} = 1$ and $p_{\Lambda_b} = 0$ then
    $cost_{\Lambda_b} \leftarrow \varphi_{\Lambda_b} r_d$
else
    $r_{\mathrm{\Lambda_{bR},protection,new}} \leftarrow r_{\mathrm{\Lambda_{bR},protection}}$
    if $X[i_{\Lambda_b}][\Xi_{\Lambda p}] + r_d > r_{\mathrm{\Lambda_{bR},protection,new}}$ then
        $r_{\mathrm{\Lambda_{bR},protection,new}} \leftarrow X[i_{\Lambda_b}][\Xi_{\Lambda p}] + r_d$
    end if
    $r_{\mathrm{\Lambda_{sE},protection,new}} \leftarrow r_{\mathrm{\Lambda_{sE},protection}}$
    for all $e \in P_{l'} : l' \in P_{path}[i_{\Lambda_p}]$ do
        if $Y[i_{\Lambda_b}][i_e] + r_d > r_{\mathrm{\Lambda_{sR},protection,new}}$ then
            $r_{\mathrm{\Lambda_{bE},protection,new}} \leftarrow Y[i_{\Lambda_b}][i_e] + r_d$
        end if
    end for
    $r_{\mathrm{\Lambda_b,protection,new}} \leftarrow \max\ (r_{\mathrm{\Lambda_{bR},protection,new}}, r_{\mathrm{\Lambda_{bE},protection,new}}($
    $cost_{\Lambda_b} \leftarrow \varphi_{\Lambda_b}(r_{\mathrm{\Lambda_b,protection,new}} - r_{\mathrm{\Lambda_b,protection}})$
end if
Calculate the lowest cost disjoint path $p_{\mathrm{protection}}$ from s to $\Lambda_b$
return $p_{\mathrm{protection}}$
```

Resilience. For PIP-Resilience since the physical redundant resources are shared, the virtual network cost is ideally not affected. The sharing of redundant resources is enabled by adding two minimization terms to the objective function, one for network and one for DC resources as given in (7.67). Since in dedicated protection (DP) MILP model [124], the resource minimization is not considered, we define a new MILP model, dedicated protection with network utilization (DP with NU), which implements dedicated protection but has additionally network resources minimization. Shared protection performance of PIP-Resilience is compared with these both models as shown in Figure 7.10. In this figure only the results of cost setting (1,1,1) are shown due to its representativeness. In this cost setting, shared protection does not cause any increase in the virtual network cost but decreases both the network and DC resource requirements. DP with NU model already causes a decrease of 9% in network resource requirements and shared protection decreases them additionally by 16% with HillClimber and 6% with kBest algorithms. DC resource requirement is decreased by around 18% with both algorithms. The same amount of DC resource saving is observed with all the cost settings. (1,A,1) has very similar results as (1,1,1). For the remaining three cost settings, virtual network cost is slightly increased by around 6-10% with shared protection. Finally, (L,1,1) and (L,A,A) allow network resource savings by 6% and 10% respectively, and with (1,1,A), 9% network resource requirement decrease with DP with NU and additionally 13% and 8% are observed with HillClimber and kBest, respectively.

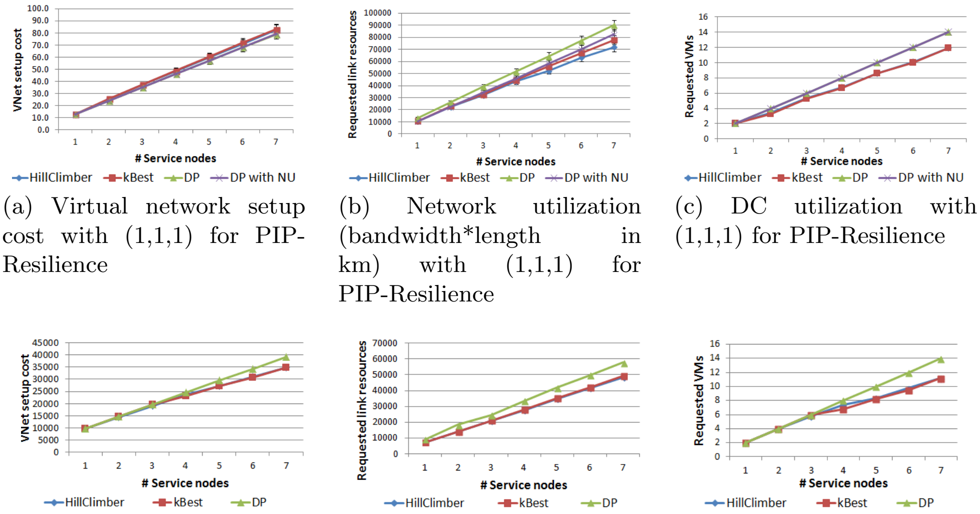
For VNO-Resilience the main savings are observed with cost setting (1,1,A), where the virtual network cost is reduced by 13%, network resource requirements by 15% and DC resource requirements by 25% as shown in Figure 7.10. (1,1,1) and (1,A,1) don't offer any cost or DC resource savings but reduce the required network resources by around 11%, and (L,1,1) and (L,A,A) allow 14% DC resource saving, however, no significant cost or network resource savings.

For VNO-Resilience both algorithms result in similar values, where for PIP-Resilience HillClimber shows slightly better results. Since both algorithms have a similar complexity, for PIP-Resilience it is advisable to use HillClimber, where for VNO-Resilience both can be used. For this analysis 2 dcPIPs having each 3 DCs are used to enable shared protection and diversity in both layers and number of service nodes is varied from 1 to 7 due to the number of nodes in the physical topology and scalability issues of the MILPs.

#### 7.4.3.3 Service Latency Comparison of Shared Protection over Dedicated Protection

In terms of service latency the performance difference of dedicated and shared protection models depends on the used cost setting and more specifically if the physical length of the virtual link is used as its cost. Using VNO-Resilience, for (L,1,1) and (L,A,A), where the results of (L,1,1) are shown in Figures 7.11a and 7.11b, the service latency is slightly increased due to shared protection in comparison with dedicated protection. This difference is more obvious for the worst case delay, namely the maximum delay. The latency increase is caused by using longer paths if necessary to enable the sharing of redundant virtual resources. However, using (1,1,1), (1,A,1) and (1,1,A), shared protection results in service latency gain due to the fact that in the dedicated protection models for these cost settings the virtual links are selected totally independent of their lengths. Therefore, shared protection offers a better optimization. The results for (1,1,1) are shown in Figures 7.11c and 7.11d.

For PIP-Resilience a similar result as the VNO-Resilience is observed as shown in Figure 7.12. For length based virtual link cost, shared protection causes an increase in service latency, where DP model and DP with NU have the same results. For fixed virtual link cost,



(a) Virtual network setup cost with (1,1,1) for PIP-Resilience

(b) Network utilization (bandwidth*length in km) with (1,1,1) for PIP-Resilience

(c) DC utilization with (1,1,1) for PIP-Resilience

(d) Virtual network setup cost with (1,1,A) for VNO-Resilience

(e) Network utilization (bandwidth*length in km) with (1,1,A) for VNO-Resilience

(f) DC utilization with (1,1,A) for VNO-Resilience

Figure 7.10: Performance comparison of shared protection and dedicated protection in terms of virtual network setup cost and resource utilization for selected cost settings

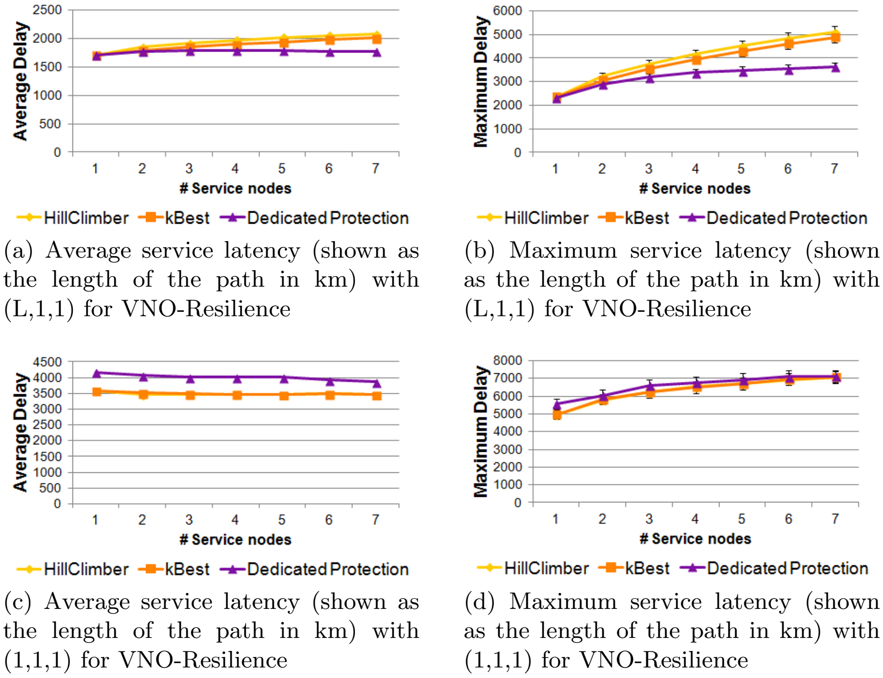
dedicated protection selects the virtual links independent of their lengths and therefore DP with NU and shared protection lower the service latency by better optimization as shown in Figure 7.12c.

#### 7.4.3.4 Virtual Network Complexity Comparison of Shared Protection over Dedicated Protection

In terms of virtual network complexity using PIP-Resilience shared protection and dedicated protection have the same results since sharing of redundant resources is realized in the physical layer, and hence, the number of the virtual links is not affected as shown in Figure 7.13d. Using VNO-Resilience, however, the results depend on the used cost setting. With (L,1,1) sharing redundant virtual resources increases the number of the required virtual links as shown in Figure 7.13a. Using (1,1,1) and the other cost settings with fixed link cost the number of virtual links remain unchanged as shown in Figure 7.13b. Finally, using (L,A,A), the number of the virtual links is slightly decreased.

#### 7.4.3.5 VNO-Resilience vs. PIP-Resilience with shared Protection

The decision of at which layer to provision resilience and shared protection depends on the priorities. In terms of network utilization VNO-Resilience is always better than PIP-Resilience by 50-300%. In terms of DC utilization the performance depends on the number of DCs per dcPIP since a PIP is limited within its domain for DR site selection and sharing. Therefore, if there are multiple PIPs with only 2 DCs in each PIP domain, VNO-Resilience can offer sharing of redundant DC resources where PIP-Resilience cannot. However, for more DCs PIP-Resilience yields lower DC resource requirements compared with VNO-Resilience. Finally, the cost performance varies with different cost settings. For the cost setting (L,1,1), VNO-Resilience is always better by 65-200% where the difference increases with increasing number of DCs and services. For (1,1,1), (L,A,A) and (1,1,A), the two models perform similarly, where VNO-Resilience starts to perform better for increasing number of DCs and services. Finally, for (1,A,1), PIP-Resilience always yields around 40% lower cost due to the higher number of virtual nodes established by VNO-Resilience.



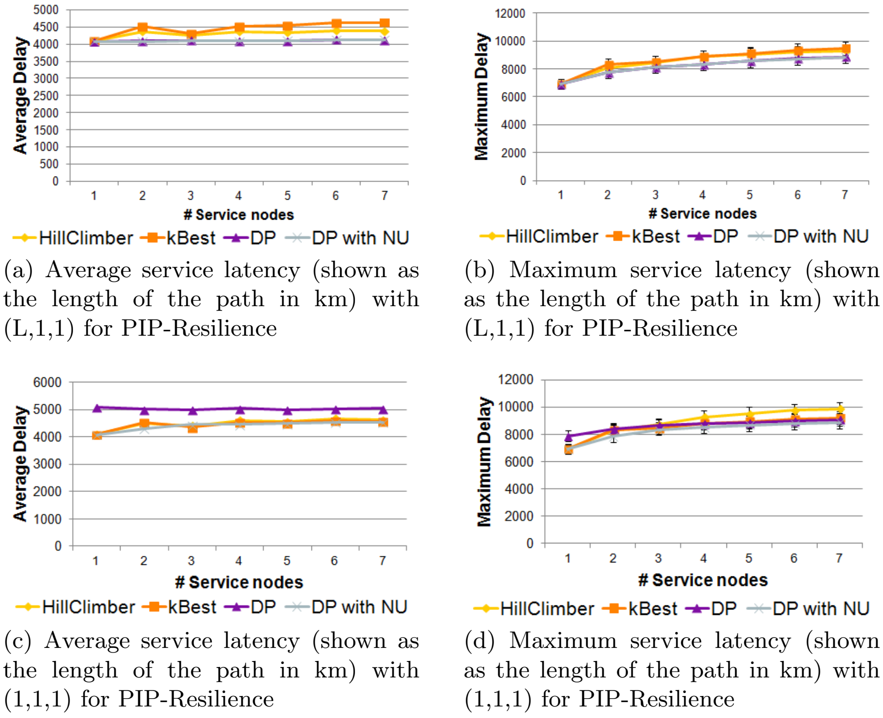
(a) Average service latency (shown as the length of the path in km) with (L,1,1) for VNO-Resilience

(b) Maximum service latency (shown as the length of the path in km) with (L,1,1) for VNO-Resilience

(c) Average service latency (shown as the length of the path in km) with (1,1,1) for VNO-Resilience

(d) Maximum service latency (shown as the length of the path in km) with (1,1,1) for VNO-Resilience

Figure 7.11: Performance comparison of shared protection and dedicated protection in terms of service latency using VNO-Resilience for selected cost settings



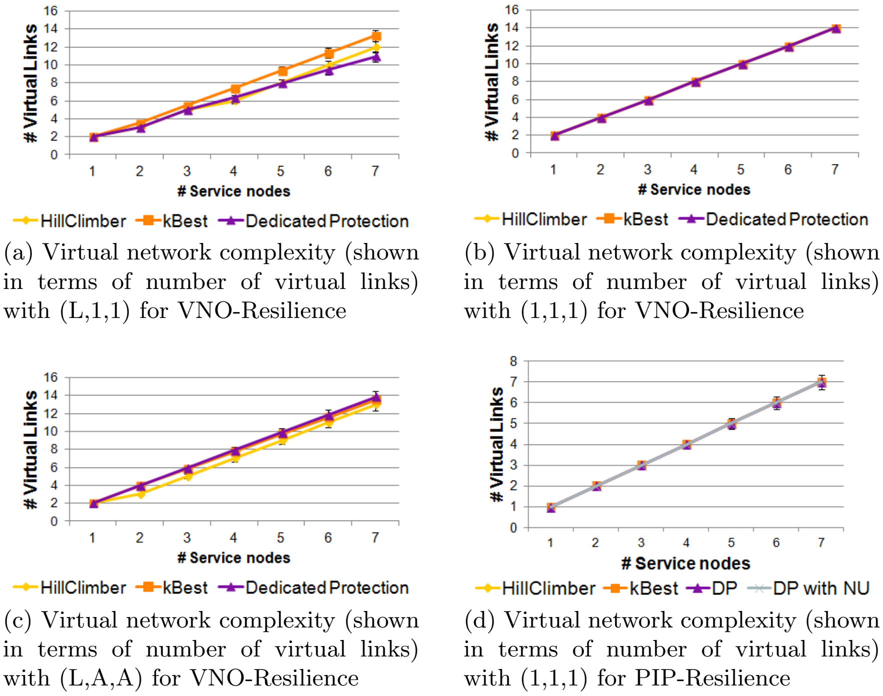
(a) Average service latency (shown as the length of the path in km) with (L,1,1) for PIP-Resilience

(b) Maximum service latency (shown as the length of the path in km) with (L,1,1) for PIP-Resilience

(c) Average service latency (shown as the length of the path in km) with (1,1,1) for PIP-Resilience

(d) Maximum service latency (shown as the length of the path in km) with (1,1,1) for PIP-Resilience

Figure 7.12: Performance comparison of shared protection and dedicated protection in terms of service latency using PIP-Resilience for selected cost settings



(a) Virtual network complexity (shown in terms of number of virtual links) with (L,1,1) for VNO-Resilience

(b) Virtual network complexity (shown in terms of number of virtual links) with (1,1,1) for VNO-Resilience

(c) Virtual network complexity (shown in terms of number of virtual links) with (L,A,A) for VNO-Resilience

(d) Virtual network complexity (shown in terms of number of virtual links) with (1,1,1) for PIP-Resilience

Figure 7.13: Performance comparison of shared protection and dedicated protection in terms of virtual network complexity for selected cost settings

In terms of service latency and virtual network setup complexity, the difference of VNO-Resilience and PIP-Resilience is not much affected by the usage of shared protection. Comparing the Figures 7.11 and 7.12, it is seen that PIP-Resilience results still in about double latency compared with VNO-Resilience. Regarding the virtual network complexity PIP-Resilience results always in fewer virtual links. This difference is mainly constant for all cost settings except for (L,1,1) because in PIP-Resilience applying shared protection does not change the results and for VNO-Resilience except for (L,1,1) the results are in the same order as dedicated protection. However, with (L,1,1), VNO-Resilience sees an increase in the number of virtual links, making PIP-Resilience with shared protection more favorable against VNO-Resilience in terms of complexity.

## 7.5 Summary

In this chapter, we show how shared protection concepts can be applied to virtual networks and how much gain they bring compared with dedicated protection. Firstly, we introduce the necessary architecture extensions and information exchange that allow the sharing of redundant virtual resources, which can be both virtualized network and IT resources. Afterwards, we present our optimization models and heuristics both for connectivity and cloud services, which incorporate shared protection. We evaluate the performance of the proposed models and algorithms via extensive simulations. This chapter provides answers to the following research questions.

**Q3.1:** Shared protection is a widely used solution in physical networks offering reduced cost and fast recovery. How can it be applied to virtual networks?

This question is answered in detail in Section 7.2. Shared protection is traditionally applied at the physical layer, where the redundant physical resources are shared. Sharing in the context of virtual networks means normally sharing of the physical substrate between different virtual networks. We apply the concept of shared protection to the virtual layer.

The main idea behind it is allowing the sharing of virtual redundant resources between different services and/or various virtual networks. Given the aims of different business roles, this concept creates a win-win situation. It lowers the virtual resource usage and hence the cost of the virtual network for a VNO. At the same time, it improves the utilization of the physical resources for a PIP, allowing it to serve more customers with its available resources.

**Q3.2:** What kind of architectural advances are necessary to enable the application of shared protection in virtual networks?

As described in detail in Section 7.2, sharing of redundant virtual resources is enabled by introducing the SHARE message as an implementation example. It is triggered by the VNO, which requests the sharing of two virtual resources, and it is realized by the PIP if desired and if possible. The complete message exchange is described in Section 7.2.2. Additionally, a certain level of information exchange is necessary to enable a VNO to determine, which resources can be potentially shared. The VNO normally lacks the topological mapping information and a PIP is unaware of the services routing. Only by exchanging a certain level of knowledge - without the disclosure of the business-critical information - redundant virtual resource sharing can be enabled.

**Q3.3:** What are the design principles for allowing the usage of shared protection concepts in the framework of network virtualization?

The main point when designing shared protection models is determining which resources can be shared depending on their physical disjointness. For the connectivity services, the working paths of two services should be physically disjoint so that their protection paths can be shared as explained in Section 7.3. For cloud services both the redundant virtual network and IT resources can be shared. Their sharing possibility depends both on the disjointness of the working paths as well as of the primary DC sites as given in Section 7.4. All details of the optimization models and heuristic algorithms for connectivity and cloud services can be found in the corresponding sections.

**Q3.4:** How much gain does shared protection bring in virtual networks?

For the connectivity services, we compare the performance of the proposed heuristics with the shared protection MILP, and show that our algorithms perform very close to the optimal solution for small virtual network instances. For larger virtual network instances, where the shared protection MILP is not scalable anymore, our algorithms offer high gains in terms of virtual network cost and physical network utilization compared with the dedicated protection MILP, which was introduced in Chapter 5.

The exact amount of the gain depends on the applied cost settings. If resilience provisioning is done in the virtual layer and virtual redundant resources are allowed to be shared by different services, a virtual network cost reduction of 30% and 15% is observed, when the link cost depends on the physical length of the virtual link and it is higher than the node cost or when the capacity dependent link cost is the dominant cost component, respectively. Moreover, for all the used cost settings a reduction in network resource requirement of 15-70% is observed indicating the gain of applying shared protection in virtual networks. When resilience and sharing are applied in the physical layer, shared protection again offers high network utilization gains ranging from 15% to 50% for different cost settings without causing any increase of the virtual network cost.

Similar to the case with connectivity services, for cloud services we also show via extensive simulations that the proposed algorithms are scalable and perform close to optimal, and that shared protection brings 10-20% improvement in cost and resource requirements both on the network and on the cloud side offering a win-win situation for the two layers.

**Q2.2:** Does virtual layer resilience bring any benefits in terms of virtual network setup cost, service latency, physical resource utilization and complexity?

Finally, the decision of applying resilience and shared protection at a certain layer depends on the actual cost setting, for which we present the best option for all the different cost settings for connectivity services in Section 7.3 and for cloud services in Section 7.4, respectively.

For connectivity services, when the link cost is dominant, VNO-Resilience performs better. It results in 90% lower virtual network cost and 35% lower network resource requirement with {L,L,1,1}, and 50% lower network resource requirement with the same virtual network cost with {$(x > 1)$,1,1,1}, compared with PIP-Resilience. For cost setting {1,1,1,1}, it is better to apply PIP-Resilience, which results in 15% lower cost and 30% lower network resource requirement compared with VNO-Resilience. For {L,L,A,A} and {1,$(x > 1)$,1,1}, there is a trade-off between cost and network utilization performance. With the former, VNO-Resilience results in 30% higher virtual network cost but 20% lower network resource requirement and with the latter in 10% reduced virtual network cost but 40% higher network resource requirement.

Using the models for cloud services, in terms of network utilization VNO-Resilience is always better than PIP-Resilience by 50-300%. In terms of DC utilization the performance depends on the number of DCs per dcPIP since a PIP is limited within its domain for DR site selection and sharing. The cost performance varies with different cost settings. For the cost setting (L,1,1), VNO-Resilience is always better by 65-200% where the difference increases with increasing number of DCs and services. For (1,A,1), PIP-Resilience always yields around 40% lower cost due to the higher number of virtual nodes established by VNO-Resilience. For the remaining cost settings the two models perform similarly. In terms of service latency and virtual network setup complexity, the difference of VNO-Resilience and PIP-Resilience is not much affected by the usage of shared protection and is comparable to the case of dedicated protection.

**Q1.5:** To cope with the possible scalability problems of the virtual network design models, what kind of heuristics can be used for resilient virtual network design?

In this chapter, the implementation details of the heuristic framework, which was briefly introduced in Chapters 5 and 6, are described in detail for the special case of resilient virtual network design with shared protection. Note that, by simply canceling the sharing property in the heuristics, they can be directly applied to the dedicated protection models, which have been presented in the previous two chapters. In this chapter, we also evaluate the performance of the proposed heuristics in terms of their scalability as well as in comparison with the optimal solution. Even for the more complicated design case of shared protection, for which the MILP models are not scalable, the proposed HillClimber and kBest algorithms perform well both in terms of computational complexity and closeness to optimality. Therefore, we conclude that the proposed heuristics are effective in solving the resilient virtual network design problem for dedicated and shared protection cases and are a very good basis for the implementation of further heuristics with additional requirements like QoS provisioning or enhanced failure coverage.

## 7.6 Statement on Author's Contributions

The Section 7.2 and the results presented in Section 7.3.1.3 of this chapter are based on [139]. Section 7.3 is giving the details of the MILP introduced in that paper and extends it by introducing the heuristics and providing a detailed performance evaluation of shared protection using the introduced algorithms. Afterwards, Section 7.4 introduces the optimization models and heuristics for the case of cloud services based on [140]. In the above mentioned publications the design of the proposed models and evaluations have been carried out by the author. The architecture extension work in [139] has been carried out together with Klaus Hoffmann and Franz Rambach.

# 8. Quality of Service (QoS) Differentiation in Virtual Networks

As introduced in the former chapters, network virtualization is seen as an enabler of the future networks, which helps to overcome the ossification of the Internet [8]. In addition, it provides a better resource control which grants the advantage of offering user or application specific virtual networks [158]. Moreover, it can also help with another limitation of the current Internet, namely that the Internet is designed for best effort services. With the increasing diversity of the services and their higher QoS needs, this limitation creates a problem as it provides the operators with restricted opportunities to distinguish themselves from their competitors [159]. Network virtualization can be one solution to this problem as it enables the operators to build their service-tailored virtual networks according to their own design criteria and the needs of the service requests [36].

Moreover, network virtualization also enables the provisioning of different services across heterogeneous infrastructure domains ensuring the requirements of each service in an end-to-end fashion, as discussed in the former chapters. Therefore, introducing service differentiation and QoS guarantees to network virtualization models has the potential to serve multiple purposes. Operation of virtual networks as isolated and service-tailored network slices enables the provisioning of new high quality services, which can have a significant impact in a service oriented market. Furthermore, it can achieve a fair usage of network resources by decoupling different service classes. Additionally, having service oriented resource allocation can relatively contribute to an increased resource utilization efficiency.

The aim of this chapter is to demonstrate the effect of introducing service differentiation to virtual networks with resilience considerations. In addition to providing protection against single link failures, where the loss of connectivity and lack of failure coverage could be very critical and even fatal to businesses [160], we will also consider complete DC outages in the second part. All in all, due to its high impact on service quality and customer satisfaction, resilience is a fundamental concern in service differentiated virtual networks and therefore will be treated as such in our models.

We extend our analysis of different layer resilience options from the other chapters with the incorporation of QoS provisioning. We consider virtual network design models in this chapter, where both the VNO and the PIP are in position to provision QoS and resilience in their respective layers. All models aim to provide resilient virtual network designs at a minimum setup cost while ensuring the QoS requirements of the services or of the virtual resources, depending on having QoS provisioning at the virtual or physical layer, respectively.

The work presented in the first part of this chapter is mainly based on our work in [106], and the second part refers to [161] regarding possible heuristic solution alternatives.

## 8.1 Related Work and Contributions

Today's Internet has been designed for best-effort services, which means that the traffic between the hosts is processed as quickly as possible but there is no guarantee for the duration or success of this delivery. Over the past twenty years, with the rapid transformation of the Internet into a commercial infrastructure, demands for service quality have rapidly developed [162]. As a result e.g. DiffServ [163] has emerged as a solution for IP networks, which enables the classification and treatment of packets according to the requirements of the traffic flow.

Moreover, QoS routing has received an intensive attention in the wireline network domain [164, 165]. The work in [166] describes QoS routing for delay sensitive multimedia applications, where they try to find paths that satisfy multiple constraints, and investigate the implications of the QoS requirements on routing metric selection. Another example is [167], where a new framework for QoS routing is proposed, which allows different levels of information to be processed at different timescales and several routing schemes that can fit into this framework are described. Finally, the work in [168] introduces a QoS architecture that ensures the delivery of different quality guarantees for different service classes in an end-to-end fashion.

QoS provisioning is not of high interest only in IP domain, but as well in other network technology domains like e.g. ad hoc networks [169, 170], where it is relatively more difficult compared with wireline networks due to constantly changing network topology. It is also an important aspect for VPNs and overlay networks. QoS-enabled Internet VPN is for example a particularly fruitful solution for corporate communications, which takes advantage of the cheap and ubiquitous Internet, but provides quality and security guarantees at the same time [171]. Such QoS enabled VPNs can be e.g. based on IPsec, where dynamic QoS treatment of traffic within a secure VPN tunnel can be provided by attaching a QoS marker to data traffic at an ingress end of the VPN tunnel [172]. Another way to provision QoS for VPNs is using MPLS, which can also provide a full range of QoS characteristics for the services [173]. The work presented in [174] discusses the dynamic allocation of virtual resources in overlay networks, which allows the adaptation to changing service requirements. The paper introduces several frameworks of reconfigurable and adaptable network operations for overlay networks in addition to a network self-knowledge approach, which provides dynamic adaptability for new services in the network.

For the case of network virtualization, which is the next step after VPNs and overlay networks, QoS provisioning is also an important topic and is investigated extensively in the literature. In [96], the virtual network requests are associated with QoS requirements, where for the virtual nodes these are e.g. the Central Processing Unit (CPU) capacity requirements and geographical location, and for a virtual link the bandwidth requirements. Another example is [175], which addresses the end-to-end QoS-aware provisioning of services in a virtual network environment, however, lacks the consideration of the different business roles in this environment. In [176], similar to [96], a discussion about QoS-aware mapping of virtual links on the physical substrate is presented, which assumes the virtual network topology defined and given as a request, i.e. as an input. However, in a virtual network environment this assumption does not hold, as a VNO needs to design its QoS-aware network with respect to the requirements of the services and the virtual resource offerings of the PIPs. Moreover, without the inclusion of these roles, a complete discussion about QoS provisioning in virtual networks cannot be provided.

Another work bringing network virtualization and QoS provisioning together is [177], which is focused on the cloud services. It deals, however, only with the migration of the services to locations near the user to improve QoS but not with the establishment of virtual network topologies, in which these services will be running. On the cloud side, another example incorporating QoS provisioning is [178], which is focused on QoS guaranteed bandwidth shifting and distribution in the networks for cloud services but discards any virtualization related aspects.

In this chapter, we propose QoS-aware virtual network design models for connectivity and cloud services, which are not restricted to virtual network embedding or just QoS routing but offer QoS provisioning in either the virtual or the physical layer. We consider as the main QoS parameter the end-to-end service latency, which is the case in many works in the literature [178, 179, 169]. We also ensure the end-to-end reliability of the services together with their QoS requirements, which is by its own a key property of today's and future networks and is also treated as an integral part of service requirements in the literature [180]. In this chapter, we also provide a discussion comparing the QoS provisioning options at the virtual and physical layers from different aspects.

## 8.2 Optimization Models with QoS Differentiation

For the design of resilient virtual networks with service differentiation, we propose three different models. The first model, as introduced in Section 8.2.1, has both VNO-level QoS and resilience provisioning, i.e. the VNO is responsible for providing service guarantees and also for acquiring protection resources to provide resilience in its domain. The second model, as presented in Section 8.2.2, has again VNO-level QoS provisioning but uses PIP-Resilience, where resilience provisioning is delegated to the PIP and the VNO rents resilient virtual resources from the PIP(s). Finally, deploying both QoS differentiation and resilience at the PIP layer, namely PIP-QoS with PIP-Resilience, is described in Section 8.2.3. When providing QoS guarantees in the physical layer, the requirements of end-to-end services are redirected to the PIP in the form of virtual resource requirements, as for the services a one-to-one mapping on the virtual links is used. Therefore, this does not let much freedom to a VNO in its virtual network design, and hence, for this case the option with VNO-Resilience is omitted.

In all of our models, we consider three QoS classes, namely the Gold, Silver and Bronze classes. Gold class is designed for critical services e.g. gaming applications, which guarantees a maximum delay of 20 ms. Silver class guarantees a maximum delay of 70 ms to e.g. real-time services like VoIP. Finally, Bronze class offers a delay of 170 ms to interactive best-effort services like web browsing. These end-to-end delay guarantees are typical examples for the backhaul or transport networks [181]. In our models, these classes are used in the classification of virtual links as well as of the services according to their end-to-end delay guarantees and requirements, respectively.

### 8.2.1 VNO-QoS with VNO-Resilience

In the VNO-QoS with VNO-Resilience model, each PIP possesses a set of $k$ virtual links available between every physical node pair, which correspond to the $k$ shortest paths between these nodes in the physical layer. We assume a certain number of virtual links from each class between all node pairs to allow the provisioning of all kind of services everywhere in the network, where the delays are mainly affected by the traffic prioritization by the PIP. The total end-to-end delay of each virtual link is calculated relative to its physical length and its ratio with the longest virtual link available in its class. In conclusion, the delay guaranteed by a virtual link is less than or equal to the delay guarantee of the corresponding QoS class, i.e. for example a virtual link with a physical path having an end-to-end delay of 14 ms is of class Gold, a link with 35 ms delay is Silver and so on. Note that a virtual link can only fall into one category, i.e. a virtual link can either be Gold, Silver or Bronze.

These virtual links are then advertised to the VNO with their corresponding QoS guarantees and the costs related to their properties. This information is used in the design of a resilient virtual network, which is optimized according to the service requests coming from the SPs and where resilience is provided in the virtual layer. Each service is routed on a primary and a protection path as in the former VNO-Resilience models. The traffic is switched to the protection path in case of failure as a 1:1 protection scheme is used. Finally, the VNO calculates a cost-optimal virtual network topology, which satisfies all the delay and resilience requirements of the services. Before going into the details of this model, the sets, parameters and variables used in all the three models are listed in the following.

- *Sets:*
  - $V$: Set of all virtual node candidates
  - $L_g$: Set of virtual links that are of Gold class
  - $L_s$: Set of virtual links that are of Silver class
  - $L_b$: Set of virtual links that are of Bronze class
  - $L$: Set of all virtual link candidates
  - $D$: Set of the requested services
  - $E_l$: Set of the end nodes of virtual link $l \in L$
  - $E$: Set of all physical links in the physical network
  - $P_l$: Set of the physical links $e \in E$, on which the virtual link $l \in L$ is mapped
  - $Z$: Set of virtual links $(j, k) \in L^2$, that share at least one physical edge or node i.e. not link and node disjoint
- *Parameters:*
  - $b_d$: Requested bandwidth for service $d \in D$
  - $d_d$: Requested end-to-end delay for service $d \in D$
  - $n_d$: Requested node resources for service $d \in D$
  - $d_l$: End-to-end delay on a virtual link $l \in L$
  - $r_{PIP}$: Cost factor of providing PIP resilience for a virtual link $l \in L$
  - $e_{cap}$: Available capacity on a physical edge $e \in E$
  - $v_{cap}$: Available resources on a virtual node $v \in V$, which are the available resources of the physical node on which $v$ is mapped
  - $\lambda_l$: Fixed setup cost for having a new link $l \in L$
  - $\theta_l$: Setup cost per unit capacity for link $l \in L$
  - $\mu_{g,v}$: Fixed setup cost for having a Gold node $v \in V$
  - $\mu_{s,v}$: Fixed setup cost for having a Silver node $v \in V$
  - $\mu_{b,v}$: Fixed setup cost for having a Bronze node $v \in V$
  - $\eta_{g,v}$: Setup cost per unit resource for a Gold node $v \in V$
  - $\eta_{s,v}$: Setup cost per unit resource for a Silver node $v \in V$
  - $\eta_{b,v}$: Setup cost per unit resource for a Bronze node $v \in V$
- *Variables*
  - $\beta_{i,d,l}$: Binary variable taking the value of 1 if the link $l \in L$ is used for the $i^{th}$ route of the demand $d \in D$, 0 otherwise
  - $\delta_{i,d,v}$: Binary variable taking the value of 1 if the node $v \in V$ is used for the $i^{th}$ route of the demand $d \in D$, 0 otherwise

- $\delta_{g,i,d,v}$: Binary variable taking the value of 1 if a node $v \in V$ is used for the $i^{th}$ route of the demand $d \in D$ and its selected class is Gold, 0 otherwise
- $\delta_{s,i,d,v}$: Binary variable taking the value of 1 if a node $v \in V$ is used for the $i^{th}$ route of the demand $d \in D$ and its selected class is Silver, 0 otherwise
- $\delta_{b,i,d,v}$: Binary variable taking the value of 1 if a node $v \in V$ is used for the $i^{th}$ route of the demand $d \in D$ and its selected class is Bronze, 0 otherwise
- $\gamma_l$: Binary variable taking the value of 1 if the link $l \in L$ is in the resulting virtual network, 0 otherwise
- $\alpha_{g,v}$: Binary variable taking the value of 1 if a node $v \in V$ is of Gold class in the resulting virtual network, 0 otherwise
- $\alpha_{s,v}$: Binary variable taking the value of 1 if a node $v \in V$ is of Silver class in the resulting virtual network, 0 otherwise
- $\alpha_{b,v}$: Binary variable taking the value of 1 if a node $v \in V$ is of Bronze class in the resulting virtual network, 0 otherwise
- $\alpha_{gs,v}$: Binary variable taking the value of 1 if a node $v \in V$ is of Gold or Silver class in the resulting virtual network, 0 otherwise
- $u_l \in [0, \infty]$: Used capacity on link $l \in L$
- $\omega_{g,v} \in [0, \infty]$: Used capacity on a Gold node $v \in V$
- $\omega_{s,v} \in [0, \infty]$: Used capacity on a Silver node $v \in V$
- $\omega_{b,v} \in [0, \infty]$: Used capacity on a Bronze node $v \in V$

The virtual network designs are formulated as MILPs. The objective function of this model aims to minimize the virtual network setup cost with respect to all different class cost values as given in (8.1). Similar to former chapters, the virtual network setup cost is defined as the sum of the individual costs of the used virtual links and nodes. The cost of each virtual resource is divided into two parts, one being the fixed cost of purchasing a new resource and the second depending on the total capacity required on that resource. The difference with the former chapters is that the cost values vary depending on the respective link or node class. The proposed terms of the cost model are considered to keep the problem's linearity, which maintains the optimization problem's simplicity.

$$\begin{aligned} \min \quad \Big( \sum_{l \in L} \lambda_l \gamma_l + \theta_l u_l \\ + \sum_{v \in V} \mu_{g,v}\alpha_{g,v} + \mu_{s,v}\alpha_{s,v} + \mu_{b,v}\alpha_{b,v} \\ + \eta_{g,v}\omega_{g,v} + \eta_{s,v}\omega_{s,v} + \eta_{b,v}\omega_{b,v} \Big) \end{aligned} \tag{8.1}$$

The constraints of the VNO-QoS with VNO-Resilience model are described in the following. The other models will be described later based on this model by highlighting the differences. For this model, since 1:1 protection in the virtual layer is applied, the number of paths, $r$, is taken as 2.

Constraint (8.2) is the non-splittable flow conservation constraint ensuring the routing of the service requests. Equation (8.3) sets the flags of the source and target nodes of a service as used. Constraint (8.4) reflects that if a virtual link is used by any service, it has to be part of the resulting virtual network topology.

$$\sum_{l: v \in S_l} \beta_{i,d,l} = \begin{cases} 1 & \text{if } v = s \text{ or } v = t \\ 2\delta_{i,d,v} & \text{otherwise} \end{cases} \quad \forall d = (s,t) \in D,\ v \in V,\ i \in \{1, .., r\} \tag{8.2}$$

$$\delta_{i,d,v} = 1 \quad \forall d = (s,t) \in D,\ v \in (s,t),\ i \in \{1, .., r\} \tag{8.3}$$

$$\gamma_l \geq \beta_{i,d,l} \quad \forall l \in L,\ d \in D,\ i \in \{1,..,r\} \tag{8.4}$$

Constraints (8.5) and (8.6)-(8.8) determine the amount of required bandwidth for a virtual link and the resources on different virtual node classes resulting from all services using this virtual resource, respectively.

$$u_l \geq \sum_{i \in \{1,..,r\}} \sum_{d \in D} \beta_{i,d,l}\, b_d \quad \forall l \in L \tag{8.5}$$

$$\omega_{g,v} \geq \sum_{i \in \{1,..,r\}} \sum_{d \in D} \delta_{g,i,d,v}\, n_d \quad \forall v \in V \tag{8.6}$$

$$\omega_{s,v} \geq \sum_{i \in \{1,..,r\}} \sum_{d \in D} \delta_{s,i,d,v}\, n_d \quad \forall v \in V \tag{8.7}$$

$$\omega_{b,v} \geq \sum_{i \in \{1,..,r\}} \sum_{d \in D} \delta_{b,i,d,v}\, n_d \quad \forall v \in V \tag{8.8}$$

Finally, the constraints given in (8.9) and (8.10) affirm the physical disjointness property for the virtual paths, where for each service all the virtual links and nodes used by the virtual working path have to be physically disjoint with those used by the protection path, except for the source and destination nodes.

$$\beta_{1,d,j} + \beta_{2,d,k} \leq 1 \quad \forall d \in D,\ (j,k) \in Z \tag{8.9}$$

$$\delta_{1,d,j} + \delta_{2,d,k} \leq 1 \quad \forall d = (s,t) \in D,\ (j,k) \in V \setminus \{s,t\} \tag{8.10}$$

The capacity constraints are defined in (8.11)-(8.14). (8.11) ensures that the total bandwidth requirement of the running services on the virtual links do not exceed the available bandwidth on the physical edges, on which these virtual links are mapped. Constraints (8.12)-(8.14) ensure the same for the node resources for each QoS class, where a virtual node can be of a single class and there is a one-to-one mapping of the nodes.

$$e_{cap} \geq \sum_{l \in L : e \in E_l} u_l \quad \forall e \in E \tag{8.11}$$

$$v_{cap} \geq \omega_{g,v} \quad \forall v \in V \tag{8.12}$$

$$v_{cap} \geq \omega_{s,v} \quad \forall v \in V \tag{8.13}$$

$$v_{cap} \geq \omega_{b,v} \quad \forall v \in V \tag{8.14}$$

Constraint (8.15) ensures the end-to-end delay guarantees on the virtual paths used by each service.

$$d_d \geq \sum_{l \in L} \beta_{i,d,l}\, d_l \quad \forall d \in D,\ i \in \{1,..,r\} \tag{8.15}$$

As mentioned before, a virtual node in one physical location is allowed to be of a single class to optimize the node resource usage. Therefore, the appropriate virtual node class should be selected to provide the necessary support for its adjacent virtual links. A Gold node can handle all types of virtual links, whereas a Silver node is able to handle both

Silver and Bronze link traffic. Finally, a Bronze node is capable of only handling Bronze link traffic.

Constraints (8.16), (8.17) and (8.18) determine if a node is assigned to a Gold or Silver class.

$$\alpha_{gs,v} \geq \alpha_{g,v} \quad \forall v \in V \tag{8.16}$$

$$\alpha_{gs,v} \geq \alpha_{s,v} \quad \forall v \in V \tag{8.17}$$

$$\alpha_{gs,v} \leq \alpha_{g,v} + \alpha_{s,v} \quad \forall v \in V \tag{8.18}$$

Constraints (8.19) and (8.20) ensure that a virtual node is of Gold class in case of having at least one adjacent Gold virtual link.

$$\alpha_{g,v} \geq \gamma_l \quad \forall (l : l \in L_g, v \in E_l),\ v \in V \tag{8.19}$$

$$\alpha_{g,v} \leq \sum_{l:l \in L_g, v \in E_l} \gamma_l \quad \forall v \in V \tag{8.20}$$

Furthermore, constraints (8.21) and (8.22) set the class of a node as Silver in case of having at least one adjacent Silver virtual link and no Gold links.

$$\alpha_{s,v} \geq \gamma_l - \alpha_{g,v} \quad \forall (l : l \in L_s, v \in E_l),\ v \in V \tag{8.21}$$

$$\alpha_{s,v} \leq \sum_{l:l \in L_s, v \in E_l} \gamma_l \quad \forall v \in V \tag{8.22}$$

Finally, constraints (8.23) and (8.24) allow a node to be of Bronze class in the presence of only Bronze adjacent virtual links and absence of any Silver or Gold adjacent link.

$$\alpha_{b,v} \geq \gamma_l - \alpha_{gs,v} \quad \forall (l : l \in L_b, v \in E_l),\ v \in V \tag{8.23}$$

$$\alpha_{b,v} \leq \sum_{l:l \in L_b, v \in E_l} \gamma_l \quad \forall v \in V \tag{8.24}$$

Three additional constraints, (8.25), (8.26) and (8.27), are needed to set the proper node class individually for the $i^{th}$ path of the service $d \in D$, which are then used on the capacity calculation of the nodes.

$$\delta_{g,i,d,v} \geq \alpha_{g,v} + \delta_{i,d,v} - 1 \quad \forall v \in V,\ d \in D,\ i \in \{1,..,r\} \tag{8.25}$$

$$\delta_{s,i,d,v} \geq \alpha_{s,v} + \delta_{i,d,v} - 1 \quad \forall v \in V,\ d \in D,\ i \in \{1,..,r\} \tag{8.26}$$

$$\delta_{b,i,d,v} \geq \alpha_{b,v} + \delta_{i,d,v} - 1 \quad \forall v \in V,\ d \in D,\ i \in \{1,..,r\} \tag{8.27}$$

### 8.2.2 VNO-QoS with PIP-Resilience

In this section, we introduce the VNO-QoS with PIP-Resilience model, where deploying QoS differentiation is still the responsibility of the VNO, whereas resilience provisioning is delegated to the PIP(s). The services are routed on a single path in the virtual layer using resilient virtual links, and resilience is provided via 1:1 protection mapping in the physical layer. The virtual links use the working path mapping in normal operation, and in case of failure, the affected traffic is rerouted in the physical layer to the protection path, which leaves the virtual topology unchanged. Compared with the previous model, all constraints are reused except that there is no further need for the disjointness constraints, (8.9) and (8.10), which are omitted in this model. Additionally, the number of virtual routes, $r$, is set to 1. Finally, a resilience premium $r_{\text{PIP}}$ is introduced into the cost model to reflect the additional cost due to physical layer resilience provisioning as given in (8.28).

$$\begin{aligned}\min \quad \Big(\sum_{l\in L}(\lambda_l\gamma_l+\theta_l u_l)r_{\text{PIP}})\\ +\sum_{v\in V}\mu_{g,v}\alpha_{g,v}+\mu_{s,v}\alpha_{s,v}+\mu_{b,v}\alpha_{b,v}\\ +\eta_{g,v}\omega_{g,v}+\eta_{s,v}\omega_{s,v}+\eta_{b,v}\omega_{b,v}\Big) \qquad (8.28)\end{aligned}$$

### 8.2.3 PIP-QoS with PIP-Resilience

In case of PIP-QoS with PIP-Resilience, realizing the QoS guarantees towards the VNO and protecting the virtual network against failures are both the responsibilities of the PIP(s). This scenario is useful for VNOs, which do not want to manage explicit QoS and resilience guarantees, and which buy end-to-end connectivity for their services from heterogeneous PIP domains. Therefore, in this model there is a one-to-one mapping between the services and the virtual links and a single hop routing is used, i.e. a service is routed on a single resilient virtual link from its source to its destination node and this link is not shared with any service from a different source or destination node. In this case, the entire model remains unchanged as in VNO-QoS with PIP-Resilience case, with the only difference being the replacement of the flow conservation constraint (8.2) with (8.29) to model the one-hop routing.

$$\sum_{l:v\in S_l}\beta_{i,d,l}=\begin{cases}1 & \text{if } v=s \text{ or } v=t\\ 0 & \text{otherwise}\end{cases} \qquad \forall d=(s,t)\in D,\ v\in V,\ i\in\{1,..,r\} \quad (8.29)$$

### 8.2.4 Performance Evaluation

In this section, we first introduce shortly the simulation framework and parameters. Afterwards, we present the simulation results with two main outcomes. We investigate the extent of service degradation in the absence of service differentiation for different service distributions, to analyze the importance of having QoS provisioning in virtual networks. Then, we compare the performance of the three aforementioned service differentiated resilient virtual network design models in terms of cost, network utilization and number of virtual links to evaluate the different layer QoS and resilience provisioning options.

#### 8.2.4.1 Simulation Parameters

For the simulations the Java Virtual Simulator Tool, which is described in Chapter 4, is used. We considered the NobelEU network [100] with 28 nodes and 41 links as the physical infrastructure. The service nodes are chosen randomly from the physical topology,

| | **Gold** | **Silver** | **Bronze** |
|---|---|---|---|
| Virtual link setup cost | 500 | 300 | 200 |
| Virtual link per unit capacity cost | 10 | 6 | 4 |
| Virtual node setup cost | 50 | 30 | 20 |
| Virtual node per unit resource cost | 10 | 6 | 4 |

Table 8.1: Simulation Parameters

while a full mesh of services with their delay requirements are assigned according to a certain service load distribution denoted as Gold/Silver/Bronze percentage, which is varied for different simulations. The services have a uniform capacity requirement. We run simulations for different number of virtual nodes, namely for 3 to 6 virtual nodes, to observe the effect of different network load. In each simulation run, the cost optimization problem is solved resulting in a virtual network topology, in which the services are routed according to their capacity and QoS requirements. The simulations are run consecutively until the results of the cost objective value are within an interval of ±5% with a confidence level of 95%. The cost parameters used by the simulation are shown in Table 8.1 and the complete list of parameters is provided in Section A.1.5. We again use a uniform demand matrix as in the other chapters. For these simulations, we assume the fixed link cost to be the dominant cost component, as the establishment of a new link of a certain class is more costly compared to increasing the capacity of an existing one and as the node class is modeled to be directly dependent on the adjacent link classes.

#### 8.2.4.2 Simulation Results and Evaluation

Our first simulation aims to investigate the importance of QoS provisioning in virtual networks and analyzes the case of its absence. For this purpose, we run simulations for computing minimum cost virtual network topologies without any QoS considerations, i.e. without applying the delay constraints for the services and aim only to minimize the total cost. We use various service load distributions, namely 40/30/30%, 60/20/20% and 20/60/20% indicating the proportion of Gold, Silver and Bronze services among all services. The presented results are an average of the simulations with 3 to 6 service nodes. Figure 8.1 shows the percentage of the services, for which the actual delay requirements are not met in the solution. It shows that 61% of the services are not satisfied with a nearly balanced service distribution. This value increases with the amount of demands that require higher guarantees. It goes up to 76% and 68% in the cases, where around two thirds of the running services inside the virtual network are of class Gold and Silver, respectively. This reflects that for a huge portion of the services, the VNO would not be able to deliver the desired quality threatening the services to be unsatisfactory, which can have serious consequences for all the involved businesses.

In the second part of the simulations, we evaluate and compare the performance of the proposed three models having QoS and resilience provisioning at different layers. Fig.8.2 demonstrates this performance comparison in terms of virtual network setup cost, amount of required network resources for the same demand set for each model and the number of the virtual links required in each case in the virtual network. For these simulations we set the PIP-Resilience cost factor $r_{PIP}$ as 2. Moreover, we assume that the service request distribution is 40% of class Gold, 30% Silver and 30% Bronze.

The simulation results show as illustrated in Figure 8.2a that having QoS provisioning at the VNO layer with PIP-Resilience offers the minimum cost for all the simulated service request amounts. The main cost difference of VNO-Resilience arises due to the cost overhead caused by providing resilience at the virtual layer, namely an increased number of virtual links as shown in Figure 8.2c, together with the fact of having a high fixed cost value for the virtual links. Similarly, Figure 8.2c shows that the PIP-QoS design results also in a much higher number of operating virtual links compared with VNO-QoS with PIP-Resilience, which explains the excess in the virtual network setup cost. The number of virtual links is also an indication of the network operation cost as it reflects the complexity and magnitude of the control plane needed to operate this virtual network. Nevertheless, delegating both QoS and resilience to the PIP leads to a more efficient network utilization as shown in Figure 8.2b. Note that for the results shown in 8.2, the resilience premium is only applied to the fixed setup cost of a virtual link. Similar to the other results presented in this thesis, if we apply it to both fixed and capacity dependent cost of a virtual link, this causes a cost increase for the PIP-Resilience models making the VNO-QoS PIP-Resilience model getting closer to VNO-QoS VNO-Resilience, however it still does not change the overall trend in the cost structure comparison.

In conclusion, QoS provisioning in virtual networks is beneficial as its absence can cause high levels of service dissatisfaction. By the decision of QoS deployment layer in virtual networks, there is a trade-off between the virtual network setup cost and resource utilization. With the used simulation parameter values, having QoS provisioning in the virtual layer is more cost-efficient, whereas resilience should be delegated to the PIP. However, in terms of network utilization, the simulation results indicate that PIP layer QoS and resilience provisioning is more beneficial.

## 8.3 QoS Differentiation in Virtual Networks with Combined Optimization

In this section, our models for designing resilient virtual networks, which provide QoS guarantees for cloud services, are introduced. Firstly, the assumptions used in this model are explained. After listing the used notation, the MILP model is given, which is divided into two parts. The first part is the basic MILP without any resilience considerations. Afterwards, we introduce the additional constraints and changes required to provision resilience either in the virtual layer, VNO-Resilience, or in the physical layer, PIP-Resilience.

### 8.3.1 Main Model Description and Assumptions

This model defines an optimization problem for creating a cost optimal resilient virtual network with QoS guarantees for a given set of cloud services. The MILP takes as an input the physical network topology with the attached DCs, the set of anycast services with their delay, bandwidth and VM specifications, set of virtual node candidates, which consist of the service nodes and the DC-network connection nodes and the virtual link candidates, where there is at least one virtual link between each service node pair and from each service

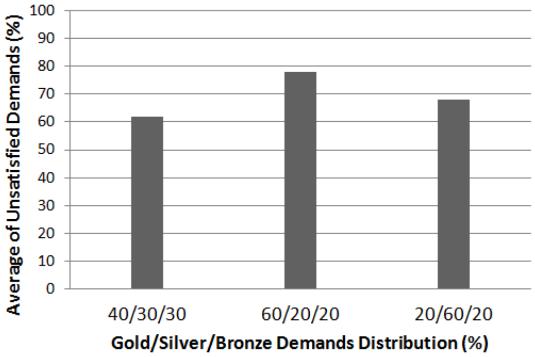
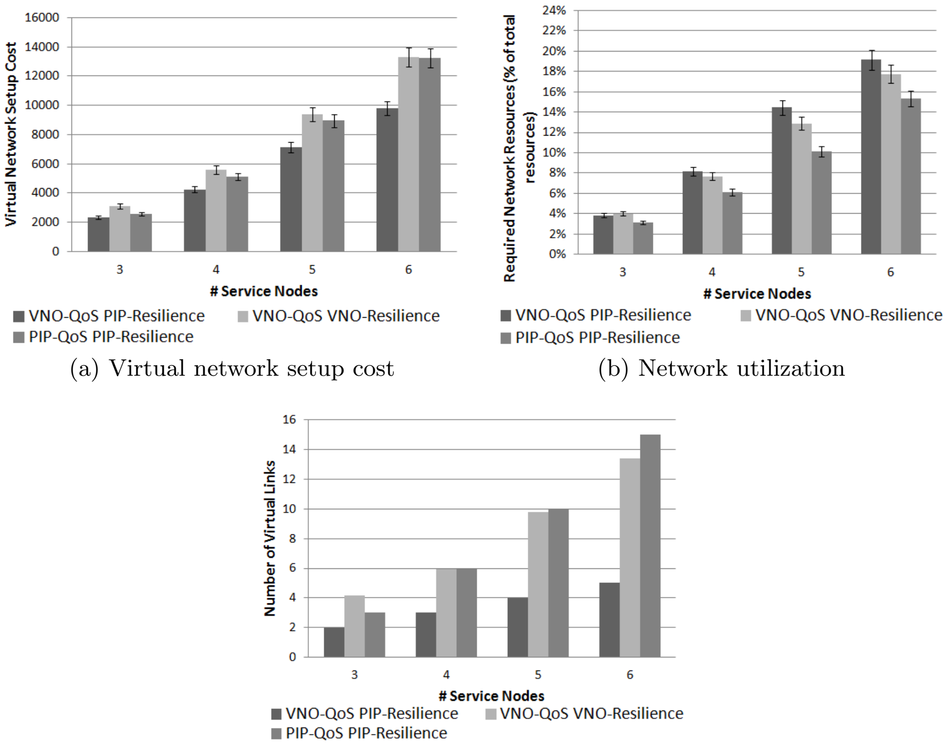


Figure 8.1: Service degradation with no QoS shown as the average of the results with 3 to 6 service nodes



(a) Virtual network setup cost

(b) Network utilization

(c) Number of virtual links

Figure 8.2: Performance comparison between the three service differentiated models with a service distribution of 40/30/30%

node to each DC node. To maintain linearity, we model different physical mappings of a virtual link as different virtual links between the same end-nodes.

The model introduces QoS differentiation for the links, nodes, VMs and services. The services have two types of QoS requirements, namely a maximum end-to-end delay and a performance criterion for the VM. The delay criterion is satisfied within the model via the delay constraints. Regarding the VM criterion, the services are divided into Gold, Silver and Bronze categories. The delay and VM criteria of the services are independent. A high computation task, which does not require live connection, might be of Gold VM class but can also be satisfied with a long network delay. However, a gaming service might require both a Gold VM class and a low network latency.

The differentiation of the virtual links is done according to their maximum end-to-end latency guarantees and are divided into Gold, Silver and Bronze delay classes. The differentiation is done by assigning a different cost value to the virtual links according to the delay class they belong to. Note that one physical edge might participate in the mappings of various virtual links with different class types.

The virtual nodes are differentiated according to their properties in terms of CPU, memory, buffer size etc. In our model we do not allow the coexistence of multiple virtual nodes of different classes belonging to the same virtual network on the same physical host due to system complexity considerations. The more virtual routers a VNO has, the higher the necessary maintenance efforts will be. Moreover, generally on one host only a limited number of instances can be initiated and in terms of resource efficiency it is favorable to keep the number of the virtual routers low. To be able to satisfy the QoS requirements, the virtual nodes are labeled within the MILP according to the highest performance required due to its ingress and egress links. The classification occurs again as Gold, Silver and Bronze classes.

The VMs in DCs are also classified as Gold, Silver and Bronze. The VM classification is done based on the performance and specifications of the corresponding VMs and these can be e.g. number of cores the VM possesses. For example Google's cloud platform provides four VM classifications based on 1, 2, 4 or 8 virtual cores.

There can be multiple VMs of different class types situated in the same DC and belonging to the same virtual network. However, one service can use only a single type of VM in one DC. During the matching process of the services to the VMs, the services' VM class type is used. A Gold service, requiring at least Gold VM performance, can be served only by a Gold class VM. A Silver service, requiring at least a Silver VM performance, can be served by either Gold or Silver VMs. Finally, a Bronze service, requiring at least a Bronze VM performance, can be served by any VM class.

In case of PIP-Resilience, each DC is connected to its protection DC via three different QoS class disjoint path pairs: Gold, Silver and Bronze respectively. The delay and QoS class of the protection path pairs is set as the property of the DC connection node.

#### 8.3.1.1 Notations

In the following the sets, parameters and variables used in the QoS-aware resilient virtual network design models for cloud services are presented.

- *Sets:*
  - $S$: Set of the services
  - $S_G$: Set of services classified as Gold (requires Gold VM)
  - $S_S$: Set of services classified as Silver (requires Silver or Gold VM)
  - $S_B$: Set of services classified as Bronze (requires Bronze, Silver or Gold VM)
  - $V$: Set of the all virtual nodes with $S \cup C = V$ and $S \cap C = \{\}$
  - $L$: Set of the virtual link candidates, where there is at least one link between all node pairs in $S$ and from each node in $S$ to all nodes in $C$
  - $L_g$: Set of virtual links that are of Gold class
  - $L_s$: Set of virtual links that are of Silver class
  - $L_b$: Set of virtual links that are of Bronze class
  - $D$: Set of the requested anycast services with all possible realizations, where $|D| = |S| * |C|$ and $d = (s, c) \in D$ with $s \in S$ and $c \in C$
  - $D_s$: Set of the requested anycast services from a service node $s \in S$ with $|D_s| = C$ and $D_s \subseteq D$
  - $E_l$: Set of the endpoints of link $l \in L$
  - $Z$: Set of virtual links $(l, k) \in L^2$, which share at least one physical edge, i.e., which are not link disjoint
  - $E$: Set of the physical links in the physical network topology
  - $P_l$: Set of the physical links $e \in E$, on which the virtual link $l \in L$ is mapped
  - $R$: Set of DC connection node pairs $(c_1, c_2) \in C^2$ with $c_1 \neq c_2$, which are located in the same availability region of the physical topology
  - $C_G$: set of DC connection nodes containing Gold class VM
  - $C_S$: set of DC connection nodes containing Silver class VM
  - $C_B$: set of DC connection nodes containing Bronze class VM
  - $C$: Set of the DC connection nodes $C = C_G \cup C_S \cup C_B$

- *Parameters:*
  - $k$: Number of the DCs to be used in total for a service with $k \in \{2, ..., |C|\}$
  - $b_d$: Requested bandwidth for the service $d \in D$

- $d_l$: Delay on the link $l \in L$
- $d_c$: Delay on the path pair connecting primary DC node $c$ to protection DC site
- $n_d$: Requested network node resources for the service $d \in D$
- $r_d$: Requested server resources for the service $d \in D$
- $\lambda_l$: Fixed setup cost for having a new link $l \in L$
- $\theta_l$: Setup cost per unit capacity for link $l \in L$
- $\mu_{g,v}$: Fixed setup cost for having a Gold node $v \in V$
- $\mu_{s,v}$: Fixed setup cost for having a Silver node $v \in V$
- $\mu_{b,v}$: Fixed setup cost for having a Bronze node $v \in V$
- $\eta_{g,v}$: Setup cost per unit resource for a Gold node $v \in V$
- $\eta_{s,v}$: Setup cost per unit resource for a Silver node $v \in V$
- $\eta_{b,v}$: Setup cost per unit resource for a Bronze node $v \in V$
- $\phi_{g,c}$: Fixed setup cost for having a Gold VM connected to node $c \in C$
- $\phi_{s,c}$: Fixed setup cost for having a Silver VM connected to node $c \in C$
- $\phi_{b,c}$: Fixed setup cost for having a Bronze VM connected to node $c \in C$
- $\varphi_{g,c}$: Setup cost per unit capacity for a Gold VM connected to node $c \in C$
- $\varphi_{s,c}$: Setup cost per unit capacity for a Silver VM connected to node $c \in C$
- $\varphi_{b,c}$: Setup cost per unit capacity for a Bronze VM connected to node $c \in C$

For PIP-Resilience

- $\varrho_g$ : Setup cost per unit capacity for Gold path pair to protection DC site
- $\varrho_s$ : Setup cost per unit capacity for Silver path pair to protection DC site
- $\varrho_b$ : Setup cost per unit capacity for Bronze path pair to protection DC site
- $d_{c,g}$ : Delay on Gold path pair connecting node $c$ to protection DC node
- $d_{c,s}$ : Delay on Silver path pair connecting node $c$ to protection DC node
- $d_{c,b}$ : Delay on Bronze path pair connecting node $c$ to protection DC node

- *Variables*
  - $a_{s,c,g}$: Binary variable taking the value of 1 if a Gold class virtual machine is placed into the DC connected to node $c \in C_G$ to satisfy the anycast demand with source s ∈ S, 0 otherwise
  - $a_{s,c,s}$: Binary variable taking the value of 1 if a Silver class virtual machine is placed into the DC connected to node $c \in C_S$ to satisfy the anycast demand with source s ∈ S, 0 otherwise
  - $a_{s,c,b}$: Binary variable taking the value of 1 if a Bronze class virtual machine is placed into the DC connected to node $c \in C_B$ to satisfy the anycast demand with source s ∈ S, 0 otherwise
  - $\beta_{d,l}$: Binary variable taking the value of 1 if the link $l \in L$ is used for the demand $d \in D$ and if demand $d = (s, c)$ is chosen as one of the realizations of the anycast service from $s$, 0 otherwise
  - $\delta_{d,v}$: Binary variable taking the value of 1 if the node $v \in V$ is used for the demand $d \in D$ and if demand $d = (s, c)$ is chosen as one of the realizations of the anycast service from $s$, 0 otherwise
  - $\delta_{g,d,v}$: Binary variable taking the value of 1 if a node $v \in V$ is used for the route of the cloud service realization $d \in D$ and its selected class is Gold, 0 otherwise
  - $\delta_{s,d,v}$: Binary variable taking the value of 1 if a node $v \in V$ is used for the route of the cloud service realization $d \in D$ and its selected class is Silver, 0 otherwise
  - $\delta_{b,d,v}$: Binary variable taking the value of 1 if a node $v \in V$ is used for the route of the cloud service realization $d \in D$ and its selected class is Bronze, 0 otherwise
  - $\gamma_l$: Binary variable taking the value of 1 if the link $l \in L$ is in the resulting virtual network, 0 otherwise

- $\alpha_{g,v}$: Binary variable taking the value of 1 if a node $v \in V$ is of Gold class in the resulting virtual network, 0 otherwise
- $\alpha_{s,v}$: Binary variable taking the value of 1 if a node $v \in V$ is of Silver class in the resulting virtual network, 0 otherwise
- $\alpha_{b,v}$: Binary variable taking the value of 1 if a node $v \in V$ is of Bronze class in the resulting virtual network, 0 otherwise
- $\alpha_{gs,v}$: Binary variable taking the value of 1 if a node $v \in V$ is of Gold or Silver class in the resulting virtual network, 0 otherwise
- $y_{c,g}$: Binary variable taking the value of 1 if a Gold class virtual machine on the DC connected to node $c \in C$ is in the resulting virtual network, 0 otherwise
- $y_{c,s}$: Binary variable taking the value of 1 if a Silver class virtual machine on the DC connected to node $c \in C$ is in the resulting virtual network, 0 otherwise
- $y_{c,b}$: Binary variable taking the value of 1 if a Bronze class virtual machine on the DC connected to node $c \in C$ is in the resulting virtual network, 0 otherwise
- $\chi_{d,g}$: Binary variable taking the value of 1 if a Gold class path pair connecting primary DC node to protection DC is selected by the service realization $d \in D$ in PIP-Resilience, 0 otherwise
- $\chi_{d,s}$: Binary variable taking the value of 1 if a Silver class path pair connecting primary DC node to protection DC is selected by the service realization $d \in D$ in PIP-Resilience, 0 otherwise
- $\chi_{d,b}$: Binary variable taking the value of 1 if a Bronze class path pair connecting primary DC node to protection DC is selected by the service realization $d \in D$ in PIP-Resilience, 0 otherwise
- $u_l \in [0, \infty]$: Used capacity on link $l \in L$
- $\omega_{g,v} \in [0, \infty]$: Used capacity on a Gold node $v \in V$
- $\omega_{s,v} \in [0, \infty]$: Used capacity on a Silver node $v \in V$
- $\omega_{b,v} \in [0, \infty]$: Used capacity on a Bronze node $v \in V$
- $z_{c,g} \in [0, \infty]$: Used capacity on a Gold class DC connected to node $c \in C_G$
- $z_{c,s} \in [0, \infty]$: Used capacity on a Silver class DC connected to node $c \in C_S$
- $z_{c,b} \in [0, \infty]$: Used capacity on a Bronze class DC connected to node $c \in C_B$

### 8.3.1.2 Objective Function

In this model, we aim to minimize the setup cost of the virtual network. This cost has three parts, namely the virtual link setup cost, virtual network node setup cost and finally the setup cost of theVMs within the DCs. These costs are shown in (8.30) - (8.38). The cost of each virtual link, node or VM is again divided into two parts. Firstly, there is a fixed cost to purchase a new link, node or VM. Secondly, there is a capacity or resource dependent cost based on the total capacity or resources required by that link, node or VM. Cost values vary depending on the respective link, node and VM class.

$$\varepsilon_l = \lambda \cdot \gamma_l + \theta_l \cdot \mu_l \quad \forall l \in L \tag{8.30}$$

$$\varepsilon_{v,g} = \mu_{g,v} \cdot \alpha_{g,v} + \eta_{g,v} \cdot \omega_{g,v} \quad \forall v \in V \tag{8.31}$$

$$\varepsilon_{v,s} = \mu_{g,s} \cdot \alpha_{g,s} + \eta_{g,s} \cdot \omega_{g,s} \quad \forall v \in V \tag{8.32}$$

$$\varepsilon_{v,b} = \mu_{g,b} \cdot \alpha_{g,b} + \eta_{g,b} \cdot \omega_{g,b} \quad \forall v \in V \tag{8.33}$$

$$\varepsilon_v = \varepsilon_{v,g} + \varepsilon_{v,s} + \varepsilon_{v,b} \tag{8.34}$$

$$\varepsilon_{c,g} = \phi_{c,g} \cdot y_{c,g} + \varphi_{c,g} \cdot z_{c,g} \quad \forall c \in C_G \tag{8.35}$$

$$\varepsilon_{c,s} = \phi_{c,s} \cdot y_{c,s} + \varphi_{c,s} \cdot z_{c,s} \quad \forall c \in C_S \tag{8.36}$$

$$\varepsilon_{c,b} = \phi_{c,b} \cdot y_{c,b} + \varphi_{c,b} \cdot z_{c,b} \quad \forall c \in C_S \tag{8.37}$$

$$\varepsilon_c = \varepsilon_{c,g} + \varepsilon_{c,s} + \varepsilon_{c,b} \tag{8.38}$$

Finally, the objective function minimizing the total cost is given in (8.39).

$$\min \quad \sum_{l \in L} \varepsilon_l + \sum_{v \in V} \varepsilon_v + \sum_{c \in C} \varepsilon_c \tag{8.39}$$

#### 8.3.1.3 Constraints

In this subsection, the constraints of the main model are introduced. Constraint (8.40) ensures that $n_{dc} \in \{2, ..., |C_G|\}$ server locations are chosen for a Gold service, which can be served by only Gold class VMs.

$$\sum_{c \in C_G} a_{s,c,g} = n_{dc} \quad \forall s \in S_G \tag{8.40}$$

Constraint (8.41) ensures that $n_{dc} \in \{2, ..., |C_G \cup C_S|\}$ server locations are chosen for a Silver service, which can be served by Gold or Silver class VMs.

$$\sum_{c \in C_G} a_{s,c,g} + \sum_{c \in C_S} a_{s,c,s} = n_{dc} \quad \forall s \in S_S \tag{8.41}$$

Constraint (8.42) ensures that $n_{dc} \in \{2, ..., |C|\}$ server locations are chosen for a Bronze service, which can be served by either Gold, Silver or Bronze class VMs.

$$\sum_{c \in C_G} a_{s,c,g} + \sum_{c \in C_S} a_{s,c,s} + \sum_{c \in C_B} a_{s,c,b} = n_{dc} \quad \forall s \in S_B \tag{8.42}$$

Constraint (8.43) states that each service can select only one VM class in a DC.

$$a_{s,c,g} + a_{s,c,s} + a_{s,c,b} \leq 1 \quad \forall s \in S, c \in C \tag{8.43}$$

Link flow constraints are given in (8.44), (8.45), (8.46) for service $s \in S_G$ , $s \in S_S$ and $s \in S_B$ respectively.

$$\sum_{l \in L: v \in E_l} \beta_{d,l} = \begin{cases} a_{s,c,g} & \text{if } v = s \text{ or } v = c \\ 2\delta_{d,v} & \text{otherwise} \end{cases} \quad \forall d = (s,c) : s \in S_G,\ v \in V \tag{8.44}$$

$$\sum_{l \in L: v \in E_l} \beta_{d,l} = \begin{cases} a_{s,c,g} + a_{s,c,s} & \text{if } v = s \text{ or } v = c \\ 2\delta_{d,v} & \text{otherwise} \end{cases} \quad \forall d = (s,c) : s \in S_S,\ v \in V \tag{8.45}$$

$$\sum_{l \in L: v \in E_l} \beta_{d,l} = \begin{cases} a_{s,c,g} + a_{s,c,s} + a_{s,c,b} & \text{if } v = s \text{ or } v = c \\ 2\delta_{d,v} & \text{otherwise} \end{cases} \quad \forall d = (s,c) : s \in S_B,\ v \in V \tag{8.46}$$

Constraint (8.47) ensures that a node $v$ is flagged as ”used” for a service $d$ if it is the source or the target node of that service realization and if this service realization is chosen for the anycast service from source $s$.

$$\delta_{d,v} = a_{s,c,g} + a_{s,c,s} + a_{s,c,b} \quad \forall d = (s,c) \in D,\ v \in (s,c) \tag{8.47}$$

Constraint (8.48) reflects that if a virtual link is used by any service, it should be part of the resulting virtual network topology.

$$\gamma_l \geq \beta_{d,l} \quad \forall l \in L,\ d \in D \tag{8.48}$$

Constraints given in (8.49), (8.50) and (8.51) reflect that if a Gold, Silver or Bronze class VM is used by any service, it should be part of the resulting virtual network topology, respectively.

$$y_{c,g} \geq a_{s,c,g} \quad \forall c \in C_G,\ s \in S \tag{8.49}$$

$$y_{c,s} \geq a_{s,c,s} \quad \forall c \in C_S,\ s \in S \tag{8.50}$$

$$y_{c,b} \geq a_{s,c,b} \quad \forall c \in C_B,\ s \in S \tag{8.51}$$

Inequalities (8.52), (8.53) and (8.54) check if a network node provides a Gold or a Silver service.

$$\alpha_{gs,v} \geq \alpha_{g,v} \quad \forall v \in V \tag{8.52}$$

$$\alpha_{gs,v} \geq \alpha_{s,v} \quad \forall v \in V \tag{8.53}$$

$$\alpha_{gs,v} \leq \alpha_{g,v} + \alpha_{s,v} \quad \forall v \in V \tag{8.54}$$

Constraints (8.55) and (8.56) ensure that a network node supports Gold class traffic in case of having at least one adjacent Gold class virtual link.

$$\alpha_{g,v} \geq \gamma_l \quad \forall (l : l \in L_g, v \in E_l),\ v \in V \tag{8.55}$$

$$\alpha_{g,v} \leq \sum_{l:l \in L_g, v \in E_l} \gamma_l \quad \forall v \in V \tag{8.56}$$

Furthermore, constraints (8.57) and (8.58) ensure that a node belongs to the Silver class in case it has at least one adjacent Silver virtual link and no Gold ones. Finally, constraints (8.59) and (8.60) ensure that a node is of Bronze class in the presence of only Bronze adjacent virtual links to it and no Silver or Gold links.

$$\alpha_{s,v} \geq \gamma_l - \alpha_{g,v} \quad \forall (l : l \in L_s, v \in E_l),\ v \in V \tag{8.57}$$

$$\alpha_{s,v} \leq \sum_{l:l \in L_s, v \in E_l} \gamma_l \quad \forall v \in V \tag{8.58}$$

$$\alpha_{b,v} \geq \gamma_l - \alpha_{gs,v} \quad \forall (l : l \in L_b, v \in E_l),\ v \in V \tag{8.59}$$

$$\alpha_{b,v} \leq \sum_{l:l \in L_b, v \in E_l} \gamma_l \quad \forall v \in V \tag{8.60}$$

Three additional constraints (8.61), (8.62) and (8.63) are needed to set the proper node class individually for the service realization $d = (s,c) \in D$.

$$\delta_{g,d,v} \geq \alpha_{g,v} + \delta_{d,v} - 1 \quad \forall v \in V,\ d \in D \tag{8.61}$$

$$\delta_{s,d,v} \geq \alpha_{s,v} + \delta_{d,v} - 1 \quad \forall v \in V,\ d \in D \tag{8.62}$$

$$\delta_{b,d,v} \geq \alpha_{b,v} + \delta_{d,v} - 1 \quad \forall v \in V,\ d \in D \tag{8.63}$$

Constraints given in (8.64) and (8.65)-(8.67) indicate the amount of capacity for a virtual link and the amount of resources on different virtual node classes resulting from the resource usage of all services, respectively.

$$u_l \geq \sum_{d \in D} \beta_{d,l}\, b_d \quad \forall l \in L \tag{8.64}$$

$$\omega_{g,v} \geq \sum_{d \in D} \delta_{g,d,v}\, n_d \quad \forall v \in V \tag{8.65}$$

$$\omega_{s,v} \geq \sum_{d \in D} \delta_{s,d,v}\, n_d \quad \forall v \in V \tag{8.66}$$

$$\omega_{b,v} \geq \sum_{d \in D} \delta_{b,d,v}\, n_d \quad \forall v \in V \tag{8.67}$$

Constraints (8.68), (8.69) and (8.70) indicate the amount of resources on different VM classes resulting from all services connected to a DC, respectively.

$$z_{c,g} \geq \sum_{s \in S: d=(s,c)} a_{s,c,g} r_d \quad \forall c \in C_G \tag{8.68}$$

$$z_{c,s} \geq \sum_{s \in S: d=(s,c)} a_{s,c,s} r_d \quad \forall c \in C_S \tag{8.69}$$

$$z_{c,b} \geq \sum_{s \in S: d=(s,c)} a_{s,c,b} r_d \quad \forall c \in C_B \tag{8.70}$$

Physical capacity and resource constraints are defined in (8.71) and (8.72)-(8.74) to make sure that the running services on the virtual links do not exceed the available physical capacity and the resources on the selected class of each virtual node do not cross the limit of the physical node resources, while (8.75) sets the limit for DC capacity.

$$e_{cap} \geq \sum_{l \in L: e \in P_l} u_l \quad \forall e \in E \tag{8.71}$$

$$v_{cap} \geq \omega_{g,v} \quad \forall v \in V \tag{8.72}$$

$$v_{cap} \geq \omega_{s,v} \quad \forall v \in V \tag{8.73}$$

$$v_{cap} \geq \omega_{b,v} \quad \forall v \in V \tag{8.74}$$

$$z_{cap} \geq z_{c,g} + z_{c,s} + z_{c,b} \quad \forall c \in C \tag{8.75}$$

### 8.3.2 VNO-QoS with VNO-Resilience

For VNO-Resilience, the services are routed in the virtual layer to $n_{dc}$ different server locations. We take $n_{dc} = 2$ as a typical number providing 1:1 protection. Both the servers and the paths leading to these servers have to be physically disjoint, such that in case of a failure at the primary site, the protection site can be used by re-routing the service there. This mechanism offers protection both against server and network failures. The additional constraints needed for this model compared with the main model are introduced in the following.

Constraint (8.76) ensures the end-to-end delay guarantees on the virtual paths chosen for each service request.

$$d_d \geq \sum_{l \in L} \beta_{d,l}\, d_l \quad \forall d \in D \tag{8.76}$$

The constraints given in (8.77) and (8.78) affirm the physical disjointness property, where for each service all the virtual links and nodes used by the virtual working path have to be physically disjoint with those used by the protection path.

$$\beta_{d_1,l} + \beta_{d_2,k} \leq 1 \quad \forall s \in S,\ (d_1, d_2) \in D_s^2, d_1 \neq d_2,\ (l,k) \in N \tag{8.77}$$

$$\delta_{d_1,v} + \delta_{d_2,v} \leq 1 \quad \forall s \in S,\ (d_1, d_2) \in D_s^2, d_1 \neq d_2,\ v \in (V \setminus \{s\})^2 \tag{8.78}$$

Constraints (8.79)-(8.81) ensure the DC region disjointness for different service classes such that the primary and protection VMs are not placed in two DCs in the same region.

$$a_{s,c_1,g} + a_{s,c_2,g} \leq 1 \quad \forall s \in S_G, (c_1, c_2) \in R \tag{8.79}$$

$$(a_{s,c_1,g} + a_{s,c_1,s}) + (a_{s,c_2,g} + a_{s,c_2,s}) \leq 1 \quad \forall s \in S_S, (c_1, c_2) \in R \tag{8.80}$$

$$(a_{s,c_1,g} + a_{s,c_1,s} + a_{s,c_1,b}) + (a_{s,c_2,g} + a_{s,c_2,s} + a_{s,c_2,b}) \leq 1 \quad \forall s \in S_B, (c_1, c_2) \in R \tag{8.81}$$

### 8.3.3 VNO-QoS with PIP-Resilience

In this model, QoS differentiation is still the role of the VNO, whereas resilience is provided by the PIP(s), and the services are routed on a single path in the virtual network to a single DC location. Resilience is provided by the PIP(s) as 1:1 protection in the physical substrate both for the network and cloud domains. The PIP(s) advertise resilient virtual links, which are mapped on a pair of physical working and protection paths. Moreover, in case of a DC outage, the traffic is routed from the failed primary DC site to the protection site in the physical layer. This re-routing and the protection DC are transparent to the VNO.

For PIP-Resilience in (8.40)-(8.42) the $n_{dc}$ value is set as 1. Since only one DC has to be selected, (8.43) is not required. Moreover, a resilience premium $r_{PIP}$ is introduced into the cost model to reflect the expenses due to resilience provisioning in the physical layer as given in (8.82).

$$\varepsilon_{l,PIP} = (\lambda \gamma_l + \theta_l \mu_l)\, r_{PIP} \quad \forall l \in L \tag{8.82}$$

In the case of PIP-Resilience each DC node is connected to its protection DC site over one of the three QoS class path pairs - Gold, Silver or Bronze. The constraint in (8.83) ensures that only one of these paths can be selected for a single service realization. If a service realization selects a certain DC site then one of the three path pairs connecting the primary DC node to the protection DC node has to be selected, and this is given in (8.84).

$$\chi_{d,g} + \chi_{d,s} + \chi_{d,b} \leq 1 \quad \forall d = (s,c) \in D \tag{8.83}$$

$$\chi_{d,g} + \chi_{d,s} + \chi_{d,b} \geq a_{s,c,g} + a_{s,c,s} + a_{s,c,b} \quad \forall d = (s,c) \in D \tag{8.84}$$

For the case of PIP-Resilience the end-to-end delay guarantee includes both the path of the service in the virtual network and the protection path pair connecting the primary DC node $c$ to the protection DC as given in (8.85).

$$d_d \geq \sum_{l \in L} \beta_{i,d,l}\, d_l + \chi_{d,g}\, d_{c,g} + \chi_{d,s}\, d_{c,s} + \chi_{d,b}\, d_{c,b} \quad \forall d \in D \tag{8.85}$$

The cost of using the protection path pair between the two DC nodes is calculated in (8.86). An extra term in the objective function is needed to select the minimum cost protection path pair between the two DC nodes that also satisfies the delay requirement of the service. The new objective function for the PIP-Resilience case is given in (8.87).

$$\varepsilon_d = \chi_{d,g} b_d \varrho_g + \chi_{d,s} b_d \varrho_s + \chi_{d,b} b_d \varrho_b \tag{8.86}$$

$$\min \quad \sum_{l \in L} \varepsilon_{l,PIP} + \sum_{v \in V} \varepsilon_v + \sum_{c \in C} \varepsilon_c + \sum_{d \in D} \varepsilon_d \tag{8.87}$$

### 8.3.4 PIP-QoS with PIP-Resilience

In this model, similar to the case with connectivity services, both QoS and resilience provisioning are the responsibility of the PIP(s) and a one-hop routing is utilized. Therefore, this model is the same as the VNO-QoS with PIP-Resilience with the difference that in the flow conservation constraints given in (8.44)-(8.46), instead of using the node indicator, the flow is forced to be 0 if it is not a direct connection between the end-points of a service.

### 8.3.5 Performance Evaluation

The aim of this section is evaluating the performance of the proposed MILP models, providing some example results, which can serve as a basis for future heuristics, and discussing shortly possible heuristic methods, which can offer scalable solutions.

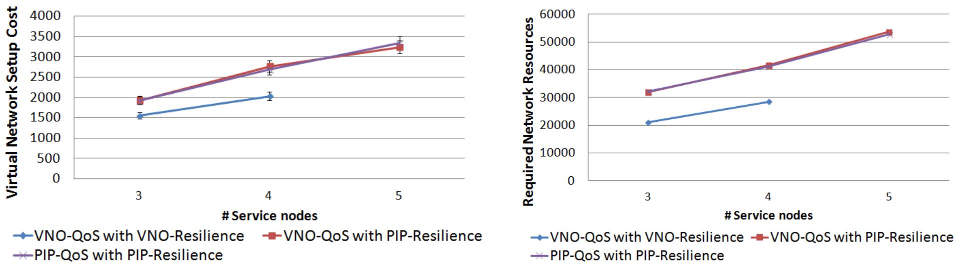
The models for cloud services that are presented in this chapter provide for the first time in the literature the methodology for incorporating QoS considerations into the simultaneous virtual network design and service routing problem. Together with the QoS constraints, which are the bandwidth and delay constraints, and the resilience requirements, this virtual network design problem is however a very hard problem. Already, the capacitated single-path flow allocation problem is NP-complete [2], which is a sub-problem for our case. More generally, already routing problems with two metrics are NP-complete [182] and the authors of [183] prove that a problem having the routing metrics delay, cost and bandwidth is NP-complete. The idea of NP-completeness is based on [184] and it means that if a problem is NP-complete then there is no known polynomial time algorithm to solve this problem.

Due to the complexity of the problem, the MILP provides optimal solutions within a feasible time on a computer with 16 cores and 60GB RAM only for 4-5 service source nodes with 4 DC locations. In this section, first we will present our results for up to 5 service source nodes and 10 services, which can be used as a good basis in evaluating the performance of future heuristics. Afterwards, we will discuss shortly possible heuristic solutions.

For the simulations, the Java Virtual Network Simulator has been used. The simulations are done using the NobelEU [100] topology. Due to the scalability issues, instead of having a simulation loop with random DC placement, we pre-select the DC locations. For this simulation, the sites are selected as London, Warsaw, Madrid and Rome on the used topology. Moreover, compared with the model for connectivity services, the delay requirements have to be updated since for cloud services the physical layer resilience uses a re-routing of the services from the primary to protection DC site, and hence, depending on the placement of the DCs, fulfilling a delay requirement as low as 20 ms is not possible for PIP-Resilience. This is one of the first important results showing that for services requiring such low end-to-end latencies both on working and protection paths, PIP-Resilience methods cannot be applied at all. Finally, the QoS class assignment method for the virtual links has been also changed to be based on the difference with the length of the shortest path between the same end-nodes, and physical node delay has been fixed to 1 ms as opposed to the unicast case, where it depends on the service class it serves. These changes has been done to support the heuristic presented in [161], which requires these simplifications as it is using a link-flow based model. A list of all parameter values is provided in Section A.1.6. As in the previous section, the fixed link cost is assumed to be the dominant cost component due to the link based QoS class assignment. The same service distribution is used as in the unicast case, which offers a rather uniform distribution of the service classes with a slight emphasis on Gold services, thus providing a balanced basis for the evaluation.

Figure 8.3 presents the virtual network setup cost and network utilization comparisons of the three models. The comparison is shown until 5 service nodes as the VNO-QoS with VNO-Resilience model runs until 4 service source nodes. Even though the results are obtained for small virtual network instances, they provide a first insight into the QoS and resilience provisioning layer problem. For the used parameter settings, in terms of both virtual network setup cost and the amount of required network resources, having both resilience and QoS provisioning in the virtual layer is better than its physical layer counterparts. The reason for this is the resilience routing drawback of PIP-Resilience, where the services cannot be re-routed from the source node but only from the primary DC site to the protection site. Moreover, the shown results are computed for a single dcPIP. Increasing the number of dcPIPs would also increase the difference between the VNO-Resilience and PIP-Resilience models. Finally, since a fixed cost value is used for the virtual links, as explained in Chapter 6, the optimal result of the MILP is connecting the source nodes and the DC sites directly on single-hop paths, where for VNO-Resilience two such connections are required per service. Therefore, VNO-QoS with VNO-Resilience model results in double number of virtual links compared with the other two models. Moreover, this routing structure is also the reason for the PIP-QoS with PIP-Resilience and VNO-QoS with PIP-Resilience having the same results in terms of virtual network setup cost, network utilization and number fo virtual links.

To overcome the scalability problem, a column-generation approach has been proposed in [161], and preliminary results are shown for a further simplified model. The column generation method [185] works by dividing the main problem into two sub-problems; a restricted master problem and a pricing problem. A restricted list of initial input candidates are given to the master problem, which provides the scalability of the model. Once the master problem is solved, the information from that solution is used in the objective function of the pricing problem, whose task is to find new input candidates to the master problem. They run alternately until no further input candidates to the master problem can be found, which improve the overall objective function. An example of this method is called path generation method, where the initial input is a sub-set of the candidate



(a) Virtual network setup cost

(b) Network utilization (Bandwidth * physical link length)

Figure 8.3: Performance comparison between the three service differentiated models with a service distribution of 40/30/30% for cloud services

paths for the routing problem, which are then updated in each iteration [2]. Even though this method is generally scalable, it does not guarantee any optimality. Therefore, the optimality gap of the proposed algorithm in [161] with the MILP models should be evaluated as a future work. Moreover, we have shown in Chapter 7 that the HillClimber and kBest heuristics perform well for the resilient virtual network design problem in terms of both scalability and optimality. Therefore, another promising solution is adapting those heuristics to incorporate the QoS guarantees for the services.

## 8.4 Summary

This chapter presents a thorough study of a service oriented network virtualization environment. We introduce three novel virtual network design models for both connectivity and cloud services that compute a cost-efficient virtual network topology considering the services' end-to-end delay guarantees in addition to resilience provisioning. The first model offers service differentiation and resilience at the VNO layer, namely VNO-QoS with VNO-Resilience. The second design suggests deploying QoS at the VNO layer with delegating the failure protection to the PIP layer, namely VNO-QoS with PIP-Resilience. Finally, in the third model, the PIP is in charge of both QoS and resilience provisioning, namely PIP-QoS with PIP-Resilience. In this chapter, all of these models are described in detail and their performance is evaluated both for connectivity and cloud services. Moreover, this chapter provides answers to the following research questions:

**Q4.1:** A good QoS provisioning is essential for customer satisfaction. How can that be done in the virtual network architecture?

In a virtual network environment, there are two interfaces, where QoS provisioning plays an important role, namely between a VNO and SP in terms of guaranteeing a certain quality for the connectivity or cloud services and between the PIP and VNO for the QoS guarantees of the virtual resources. The former denotes e.g. the bandwidth, computational resource and end-to-end latency requirements of a service. The latter provides a guarantee to a VNO that e.g. a virtual link has a certain maximum latency as it belongs to a certain QoS class. For this purpose we introduce the concept of QoS classes to the virtual networks, where they are applied both to define the service requirements and the properties of virtual resources. Additionally, we also consider the resilience requirements of the services within our virtual network design and service routing. To evaluate the importance of QoS provisioning in virtual networks, we conduct simulations, where these constraints are discarded and the only focus is minimizing the virtual network setup cost. Our results show in this case a remarkably deteriorated service delivery quality, which points out that investing towards supporting QoS guarantees is a concrete need and their absence can be very costly for businesses.

**Q4.2:** At which layer is it better to provide QoS guarantees?

As described above, we have proposed three different options concerning QoS and resilience provisioning. For the connectivity services, extensive evaluation of the resulting service differentiated resilient virtual networks with these models shows that deploying QoS at the VNO layer while delegating resilience provisioning to the PIP layer relatively outperforms the other two models in terms of the virtual network setup cost with the used parameter settings. Nevertheless, considering PIP-QoS with PIP-Resilience results in a better network resource utilization. Supporting both service differentiation and resilience at the VNO layer results both in terms of cost and network utilization in the middle of the two other models, and hence, it can serve as a trade-off solution.

For cloud services, however, having both QoS and resilience provisioning in the virtual layer results better than the physical layer solutions both in terms of the virtual network setup cost and network utilization for the used parameter settings. The reason behind this is the resilience routing in PIP-Resilience, where the services are re-routed from the primary site to the protection site instead of re-routing at the service source node.

## 8.5 Statement on Author's Contributions

The work presented in the first part of this chapter about QoS provisioning for connectivity services is mainly based on [106]. The presented experiments have been carried out by Arsany Basta in the framework of his master's thesis under the supervision of the author. Compared with the paper, the text has been edited and the Figure 8.2b has been updated to show the results with network utilization being the simulation aim, i.e. making the results to be in a ±5% confidence interval with 95% confidence level. In this chapter, we additionally present our models with QoS provisioning in virtual networks for cloud services, which constitutes the second part of this chapter.

# 9. Failure Coverage of Different Virtual Network Design Models

As introduced in the former chapters, in our virtualization model, we consider two business roles, namely the PIP and the VNO. The PIP is the owner of the physical network assets, on which the virtual topology is deployed, while the VNO acquires virtual resources from one or more PIPs and operates a virtual network to provide connectivity for SPs. For such a architecture with multiple layers, namely physical and virtual layers, to be reliable, resilience needs to be provided against several failure types that could occur in both the physical or the virtual layers. These failures might be single or double physical link failures resulting e.g. from a fiber cut, physical node failures, virtual link or node failures, or sub-network failures. Even though multiple failures and sub-network failures have a low occurrence rate, due to their remarkable impact on the network's operation and service delivery, they need to be included into the virtual network design to ensure seamless service delivery in next generation networks. For example, even though 70% of the network failures are due to single physical link failures [51], other failure types that have a lower occurrence rate such as multiple physical or virtual link failures impose a remarkable impact on the network's operation and in case of sub-network failures, extensive services interruption in a large subset of the network is experienced as reported at Japan's tsunami of 2011 [186]. Therefore, especially large operators tend to invest in higher degrees of reliability to be able to fulfill the strong SLA requirements of their customers.

In the proposed models in this chapter, similar to the previous chapters, each business role can provide resilience in its own layer. A VNO has the advantage of being able to handle both virtual and implicitly physical failures, while a PIP requires a monitoring access to handle virtual failures. However, a PIP is faster to detect and recover physical failures since it is closer to the origin of the physical failures. In addition, a VNO could experience a cost overhead for acquiring more virtual resources for resilience purposes. Therefore, it is important both to determine which failures are detectable and recoverable at each layer as well as deciding at which layer to apply resilience depending on the protection and cost requirements.

In this chapter, we first provide an analysis of the possible failures in a virtual network environment. These failures are classified according to the layer, at which they are detectable and recoverable. Afterwards, we propose virtual network design models that expand the failure coverage against several failure types in both physical and virtual layers. The design aim is finding a virtual network topology with minimum cost that offers the requested level of resilience provisioning. The basic model is introduced in Chapter 5, which considers only single link and node failures. In this chapter, it is extended to protect against double link failures, virtual node failures in addition to sub-network failures, where each design is modeled as a MILP with an objective of minimizing the virtual network setup cost. Finally, the cost related to each protection scheme is evaluated and a cost vs. failure coverage trade-off analysis is presented.

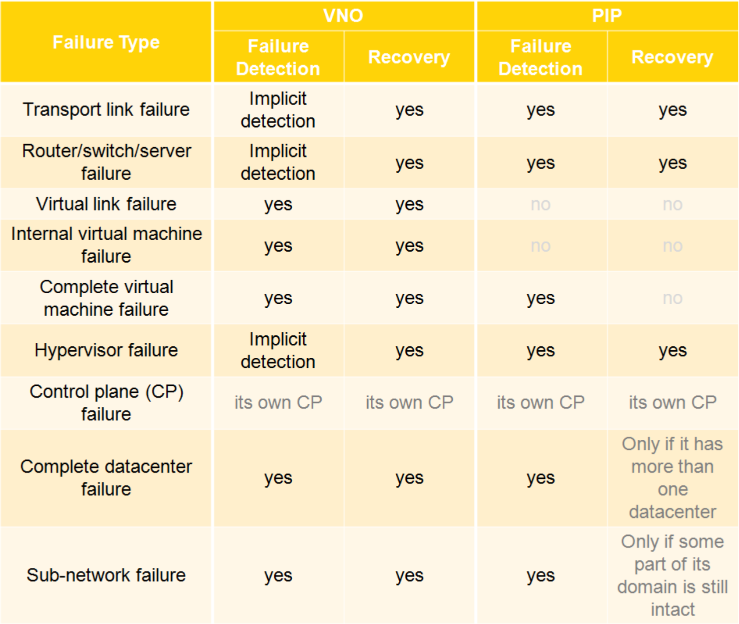
The optimization model and simulation results in this chapter are based on our publications [187] and [72].

## 9.1 Related Work and Contributions

Resilience has been and is a crucial part of the network design. Numerous studies have been conducted considering resilience provisioning in the physical layer, as in the optical and MPLS domains [188, 189]. Even though single physical link failures are the most common failure types in networks [51], and consequently, most network resilience literature deals with protection mechanisms against this failure type, there is also extensive literature on enhanced resilience design. Increased resilience also increases the cost and therefore the literature focuses mainly on reaching a certain resilience level with a minimum cost increase. Protection against double-link failures can be provided by either designing the network to cover these type of failures at a minimum cost [190, 191, 192] or enhancing a model, which guarantees single link failure protection, to cover as many double-link failures as possible [193]. Protection against double-link failures is motivated by the fact that the single link failures are common and even though the recovery is realized in a few milliseconds, repairing the failed link will last in the range of hours. Therefore, it is probable that another link fails during this time interval causing a double-link failure [194].

Another failure type, which is less frequent but has a larger impact, is a geographical failure or a sub-network failure. They can occur due to natural disasters like earthquakes, tsunamis, hurricanes etc. or due to e.g. terrorist attacks. In such cases, it is very important that the communication networks survive the failure and the connectivity and data access services are up and running. To realize this aim the literature offers solutions for the network resilience side for disaster recovery [195], as well as IT resilience mechanisms like asynchronous mirroring [196] and virtual appliance migration [197], enabling the usage of a pre-configured backup side after the failure. Finally, Habib et al. present an optimization model for anycast network design, which considers both the network and DC disaster recovery [198].

There have been some investigations towards increasing the virtual networks' reliability as the work presented in [199] and [200]. However, these studies exclude virtual failures and offer protection in the physical layer only. The papers [201] and [202] deal with the problem of resilient virtual network mapping in case of single physical link and node failures, respectively. The work by Gu et al. [203] covers the virtual network mapping problem with disaster recovery by realizing protection mapping in different geographical regions. Finally, the work in [204] deals with post failure restoration by re-mapping certain parts of the virtual networks to provide disaster recovery. All of these works, however, fall short in answering the question of how a cost-efficient virtual network can be designed by a VNO for different levels of resilience since they assume the virtual network topology to be pre-given and deal solely with the resilient mapping problem. Our models, introduced in Chapters 5 and 6, answer the virtual network design question, however they provide protection only against single link or node failures and do not consider an extended failure coverage.



| Failure Type | VNO | | PIP | |
|---|---|---|---|---|
| | Failure Detection | Recovery | Failure Detection | Recovery |
| Transport link failure | Implicit detection | yes | yes | yes |
| Router/switch/server failure | Implicit detection | yes | yes | yes |
| Virtual link failure | yes | yes | no | no |
| Internal virtual machine failure | yes | yes | no | no |
| Complete virtual machine failure | yes | yes | yes | no |
| Hypervisor failure | Implicit detection | yes | yes | yes |
| Control plane (CP) failure | its own CP | its own CP | its own CP | its own CP |
| Complete datacenter failure | yes | yes | yes | Only if it has more than one datacenter |
| Sub-network failure | yes | yes | yes | Only if some part of its domain is still intact |

Figure 9.1: List of possible failures in a virtual network environment and at which layer they are detectable and recoverable

In this chapter, we first introduce the different failure types, which might occur in a virtual network environment and classify them according to the layer they can be detected and recovered at. Afterwards, we present the enhanced optimization model, which offers resilience at various levels like protection against double-link failures, virtual layer failures and sub-network failures. Finally, we discuss what is a feasible level of resilience in terms of cost and at which layer it is better to provision it.

## 9.2 Classification of Failures in a Virtual Network Environment

When deciding on a resilience alternative, the types of potential failures are one of the main considerations. In this section, we briefly list possible hardware and software failures in a virtual network environment and then discuss which of the business roles, which were introduced in this thesis, is in the position to detect them and to recover from them.

Figure 9.1 presents a list of the different failure scenarios and Figure 9.2 illustrates a virtualized physical node and link describing where these failures occur. We start with the most common failure type in transport networks, namely physical link failures. A PIP is the owner of the physical infrastructure and can therefore detect and recover from the physical link failures. Since it is closer to the origin of the failure and since a physical link is usually shared among different virtual networks, a PIP can offer a fast and scalable recovery. A VNO is in the position of implicitly detecting a link failure, meaning that it recognizes the failing connection inside its virtual network but cannot detect its actual cause. However, it can apply recovery actions like rerouting the traffic within its virtual network. It has more flexibility due to its overview of different PIP domains while selecting the new route, however, such a recovery action must be taken by every affected VNO separately. The detection of and recovery from a physical node failure is analogous to the case of physical link failures.

In a virtual network environment, another type of link failure is a virtual link failure, signifying that the virtual link interface fails. Since the virtual interface failure is an

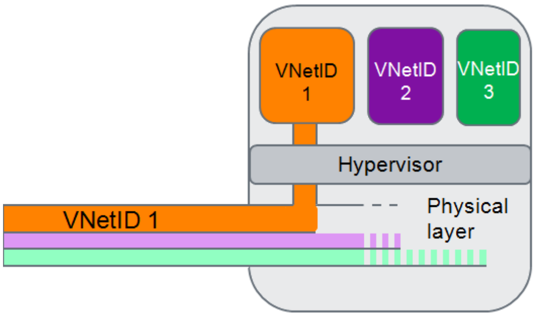


Figure 9.2: Partitioning of a network node and a network link. Failures can occur in any part of the illustrated architecture. The recovery of the parts indicated with gray color, namely the recovery of the complete physical node, of the hypervisor, of the physical link connection and of the physical link, are in the responsibility of the PIP. The resources colored in orange, violet and green, belong to different virtual networks. A problem occurring in this zone, e.g. an internal or complete failure of a virtual machine or a failure of the virtual link interface, has to be solved by the corresponding VNO.

internal failure of the virtual router, a PIP is not in the position to detect it, and hence, cannot offer recovery from it. The VNO needs to address this problem and can apply a similar recovery action like in the case of a physical link failure. Moreover, a general internal virtual machine failure at the network or server side like a software problem, buffer overflow, etc. can be only detected and solved at the VNO side, except for a complete virtual machine failure that can be as well recognized by a PIP. Still, normally it is in the responsibility of a VNO to restart its virtual machines and take the necessary recovery actions.

In case of a hypervisor failure, which is similar to a physical link/node failure, both roles can detect the failure and recover from it; however, to solve the cause of the problem is the responsibility of the PIP. In case of a control plane failure, each layer can detect the problems within its own control plane and can react on them only. However, since in that case the data plane continues to work and hence a fast recovery is not required, we do not go into more details of this problem.

Finally, protection against complete datacenter failures or sub-network failures, shortly disaster recovery, can be provided by both business roles. In both of these failure types, where a part of a physical domain or a complete domain is affected, the PIPs might have a disadvantageous position compared with the VNOs, which have an overview of different PIP domains. For example, if a PIP only possesses a single datacenter or if the complete PIP domain goes down, it has no chance of offering any recovery for the failed services. However, a VNO can make use of the other available network and cloud domains and can even have a solid disaster recovery strategy by selecting its resources in advance from disjoint physical domains or availability regions. Availability regions are ideally pre-determined such that a failure in one region does not affect the other regions.

In conclusion, recovery against physical failures can be provided by both business roles, where problems occurring within the virtual layer can be only detected and reacted on by the VNOs. Therefore, for the physical failure protection, one can choose to provide resilience in the physical or in the virtual layer or using a combination of both. A recovery strategy in the virtual layer requires reserving redundant virtual resources in advance or requesting them in case of failure depending on the level of protection required, increasing the cost and level of necessary network management knowledge at the VNO. The PIP layer can cope better with physical failures but is restricted in terms of accessing the resources of other domains. Therefore, at which layer to provision resilience is not a trivial problem and it is discussed further in the next section.

## 9.3 Virtual Network Design with Enhanced Failure Coverage

In this section, the extensions to the basic model as presented in Chapter 5 are introduced, which are required for enhancing the failure coverage. First, the basic model design principles are recalled and then the necessary extensions for each protection type are shown and explained.

### 9.3.1 Protection against Physical/Virtual Single Link and Physical Node Failures

The model for both VNO and PIP level protection against physical single link/node failures is introduced in Chapter 5. In VNO-Resilience, each virtual link is mapped on a single physical path, while each service is routed on two physically disjoint virtual routes. In PIP-Resilience, each virtual link is mapped on two disjoint physical paths and each service is routed on a single route in the virtual layer. Note that VNO-Resilience also provides protection against single virtual link failures since physical disjointness includes virtual disjointness.

### 9.3.2 Protection against Double Link Failures

Similarly, in VNO-Resilience each virtual link is again mapped on a single physical path, while each service is now routed on three physically disjoint virtual routes according to the constraints (9.1), (9.2) and (9.3). In PIP-Resilience, each virtual link is mapped on three disjoint physical paths, while each service is routed on a single virtual path. Resilience against double virtual link failures, i.e. failures of the virtual interfaces, is implicitly provided, and as in the case of single link failures, it can be provided solely by the VNO.

$$\beta_{1,d,j} + \beta_{2,d,k} \leq 1 \quad \forall d \in D,\ (j,k) \in Z \tag{9.1}$$

$$\beta_{1,d,j} + \beta_{3,d,k} \leq 1 \quad \forall d \in D,\ (j,k) \in Z \tag{9.2}$$

$$\beta_{2,d,j} + \beta_{3,d,k} \leq 1 \quad \forall d \in D,\ (j,k) \in Z \tag{9.3}$$

The set $Z$ is the set of virtual links $(j,k) \in L^2$, that share at least one physical edge or node and $\beta_{i,d,l}$ is a binary variable taking the value of 1 if the $i^{th}$ route of demand $d \in D$ uses the virtual link $l \in L$, and 0 otherwise.

### 9.3.3 Protection against Virtual Node failures

Protection against virtual node failures is considered to be only provided by the VNO. It denotes the failure of the virtual machine operating on the physical node. Therefore, the VNO acquires a backup virtual machine that would operate in case the primary virtual machine fails. This is reflected as a cost factor added to the virtual node setup cost as demonstrated in the virtual network cost minimization expression (9.4), which constitutes of the link and node setup costs.

$$\min \quad \Big( \sum_{l \in L} \lambda_l \gamma_l + \sum_{l \in L} \theta_l u_l + \sum_{v \in V} \mu_v \alpha_v r_{\text{VNO}} + \sum_{v \in V} \eta_v \omega_v \Big) \tag{9.4}$$

The binary variables $\gamma_l$ and $\alpha_v$ take the value of 1 if the virtual link $l$ or node $v$ are part of the resulting virtual network, respectively. $u_l$ and $\omega_v$ provide the information of the requested capacity on the specified link or node and $\lambda_l$, $\theta_l$, $\mu_v$ and $\eta_v$ are the cost parameters of the fixed and capacity dependent costs of the links and the nodes, respectively. Finally, $r_{\text{VNO}}$ represents the additional cost factor for providing virtual node protection at the VNO and is taken as 2 in the simulations.

### 9.3.4 Protection against Sub-Network Failures

Sub-network failures occur due to natural disasters e.g. hurricanes or tsunamis and they cause all the equipment in a certain region to fail. Protection against such failures is provided by routing the working and protection paths of each service in different availability regions, where a failure in one availability region does not affect the operation of the other regions. Therefore, the PIP would map each virtual link on two region disjoint physical paths, while the VNO would route each service on two physically region disjoint routes in the virtual layer using the constraint (9.5).

$$\beta_{1,d,j} + \beta_{2,d,k} \leq 1 \quad \forall d \in D,\ (j,k) \in Z' \tag{9.5}$$

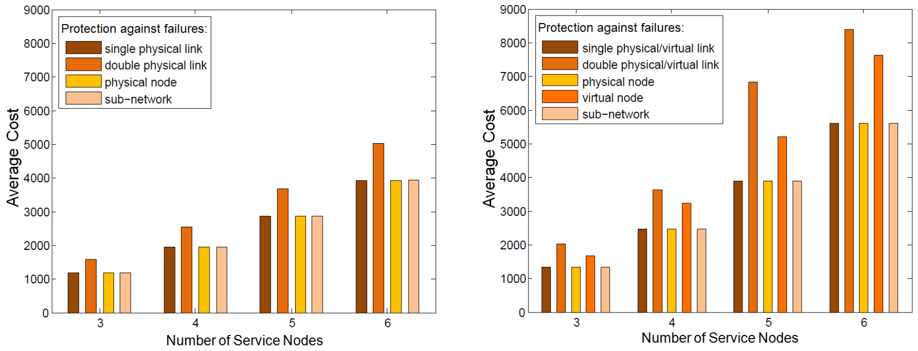
The set $Z'$ is the set of virtual links $(j,k) \in L^2$, that share at least one physical node from the same region except for the source and destination nodes i.e. not region disjoint.

### 9.3.5 Performance Evaluation

In this section, we give an insight on how the virtual network cost changes with increased protection levels to help operators in future in their decision on a feasible level of resilience provisioning. Figure 9.3 shows a virtual network setup cost comparison for different levels of protection both with PIP-Resilience and VNO-Resilience. In our analysis, we define the setup cost of a virtual network as the summation of virtual link, node and virtual machine (if used) costs. Each of these cost components consist of a certain fixed cost value signifying the cost of setting up this virtual element and a capacity dependent cost value per unit capacity requested on the virtual element. For this evaluation the example cost factors are chosen such that fixed cost components are higher than the capacity dependent cost components and link cost is the dominant cost factor. The reason for this was the assumption that the initial setup of a virtual element can be more costly than increasing its capacity incrementally.

The simulations are performed using the Java Virtual Network Simulator as introduced in Chapter 4. The NobelEU network [100] was used as the physical infrastructure and it was modified as to be 3-node-connected to support double link failures coverage as well. The simulation results are within an interval of ±5% with a confidence level of 95%. We simulated both VNO-Resilience and PIP-Resilience models for all the introduced failure types over random virtual nodes ranging in number from 3 to 6. We choose as link (node) cost parameters 200 (4) and 20 (4) for the fixed and capacity-dependent setup costs, respectively, as the links have a higher degree of freedom in optimization than the nodes. A short list of simulation parameters is given in Section A.1.7. The simulation parameters are the same as for Chapter 5 except for the extended physical topology as mentioned before.

Under the given assumptions, it is shown that for all kind of failures PIP-Resilience results in a lower cost value than VNO-Resilience. The most interesting result is that providing resilience against single link, node or sub-network failures has almost the same cost to an operator. In a sub-network failure, it is assumed all the links and nodes in a certain availability region fail simultaneously due to e.g. a disaster. Thus, an intelligent virtual



(a) PIP-Resilience

(b) VNO-Resilience

Figure 9.3: Failure coverage vs. virtual network setup cost for (a) PIP-Resilience and (b) VNO-Resilience. Protection against single link/node failures and sub-network failures is realized by using two link/node or sub-network disjoint paths for the routing of the services or for the mapping of the virtual links, respectively. In a sub-network failure, all the links and nodes in that sub-network are assumed to be failed. In case of protection against double link failures, three link disjoint paths are utilized. Number of the service nodes signifies the different source node locations of the services. For this simulation, a fixed link cost is used and the link cost is dominant compared with the node cost. Each virtual element has a cost component for establishing this virtual element and a second part linearly depending on the requested capacity on this element. The used setup and capacity dependent values for the link/node cost are 200/4 and 20/4, respectively. Due to the usage of fixed cost values, the relative cost behavior is mainly not affected by the number of service nodes.

network design enables having disaster resilience at the same cost as single link failure protection.

Moreover, if protection against double link failures is requested, namely protection against simultaneous failure of two independent links, the cost increase compared to single link failure protection is significantly lower with PIP-Resilience than with VNO-Resilience. Protection against double link failures at the PIP level requires an addition of around 46% to the cost, which is expected as each virtual link is mapped on three disjoint physical paths for double link failures protection instead of only two in all other failures.

VNO-Resilience can protect against virtual failures as well, however it experiences a higher cost for all the protection mechanisms being up to 37% more than the PIP. However, note that, failures occurring in the virtual layer can only be detected and recovered from in the virtual layer. Moreover, if VNO-Resilience is already applied to protect against other failure types, protection against virtual layer failures can be realized without any cost increase compared with protection against their physical counterparts. In other words, if protection against virtual link and node failures is already provisioned in the virtual layer, it is more cost efficient to request a non-resilient network from the PIP(s) since single physical link and node protection is implicitly provided in the virtual layer.

## 9.4 Summary

This chapter deals with the failure coverage aspect of the virtual network design. Even though multiple failures and sub-network failures have a lower occurrence rate, due to their high impact, they need to be included into the protection design. There are important questions to be answered, namely what is a feasible protection level, which failures can be detected and recovered from at which virtualization layer and at which layer one should provision resilience in the light of these findings. These points are summarized in the following answers to the research questions, which are handled by this chapter.

**Q2.3:** What kind of failures can occur in a virtual network architecture and by which layer are they detectable and recoverable?

This chapter provides a detailed list of virtual and physical layer failures, which can occur in a virtual network environment. A list of these failures together with the layers they are detectable at and recoverable is given in Figure 9.1. Some failures like virtual link and VM failures are only recoverable from at the virtual layer due to separation of the operation of the different layers. Moreover, a PIP also has difficulties to recover from DC and sub-network failures due to its domain-based limitations in recovery. However, it has the advantage of being close to the origin of the failure in case of physical failures. Therefore, the decision of what resilience layer to use depends on the required protection level as well as other factors like the cost of having this resilience.

**Q2.4:** What is a feasible level of protection in terms of the failure coverage vs. cost trade-off in a virtual network?

The most interesting finding of this chapter is that both VNO-Resilience and PIP-Resilience models can offer protection against single link, single node and sub-network failures for the same virtual network cost. In other words, having additional protection against node failures and sub-network failures does not increase the virtual network cost compared with protection against single link failures. However, if protection against double link failures or VM failures is desired, higher cost values are observed. Cost overhead for protecting against double link failures is around 46% for PIP-Resilience and 34% for VNO-Resilience, and for protection against virtual node failures up to 26% for VNO-Resilience. The decision for the protection level for these cases should be taken according to the SLA requirements and the cost benefit this additional protection level offers in terms of network operation.

**Q2.2:** Does virtual layer resilience bring any benefits in terms of virtual network setup cost, service latency, physical resource utilization and complexity?

In this chapter the focus is the cost of the virtual network since a failure coverage vs. cost trade-off analysis is carried out. For the used cost settings, which consist of a fixed and dominant link cost value, protection at the virtual layer results in a cost overhead of up to 37% compared with protection in the physical layer. However, certain failures can be only recovered from at the virtual layer. If protection against these virtual layer failures (virtual link and node failures) is already in place in the VNO-Resilience model, protection against physical layer failures (physical link and node failures) is implicitly provided without any additional cost.

## 9.5 Statement on Author's Contributions

The Section 9.3 of this chapter is based on [187]. The presented experiments have been carried out by Arsany Basta in the framework of his master's thesis within the supervision of the author. Additionally, classification of possible failures in a virtual network environment together with the analysis at which layer they can be detected and recovered from is provided and is based on [72].

# Part V

# Conclusion

# 10. Selection of the Layer for Provisioning Resilience in a Virtual Network

One of the main questions this thesis addresses is the decision about the resilience-provisioning layer, in other words, if it is more beneficial to provide resilience in the virtual or rather in the physical layer. We answer this question, as a result of performance evaluations of the proposed models and algorithms, in different chapters. This chapter presents a summary of these findings to provide an overview for the answer to the aforementioned question regarding the benefits and drawbacks observed in terms of the virtual network setup cost, service latency, resource utilization and complexity with the different resilience design alternatives.

We first start with the results of the models with connectivity services from Chapter 5. In that chapter, we propose two resilience alternatives, namely VNO-Resilience and PIP-Resilience, where protection against single link or node failures is solely provided at the virtual or physical layer, respectively. The performance evaluation of the models has been done via extensive simulations using the simulator described in Chapter 4. In terms of virtual network setup cost, the decision on the resilience model depends on the selected cost setting. We have defined six cost settings, which measure the effect of the dominance or equality of virtual link and node costs and the dependency of the virtual link cost on its physical length. Dominance of link cost causes the virtual layer resilience to be more cost-efficient, where PIP-Resilience results in around 35% excess cost. For the case of cost equality or dominance of node cost, physical layer resilience should be preferred, since VNO-Resilience results in 20-40% higher cost, depending on the influence of node cost in the total cost amount. This behavior is caused due to the fact that in VNO-Resilience, which uses protection routing and redundant resources inside the virtual network, more virtual node resources are necessary compared with PIP-Resilience. In terms of service latency, VNO-Resilience is favorable in case virtual link length is optimized with the cost, where otherwise with PIP-Resilience services need to cope with 45% higher latency. For other cost settings, no large differences are observed between the two models in terms of latency. The results for network resource requirement are aligned with the latency results. Finally, due to redundancy provisioning within the virtual layer, VNO-Resilience results in around 50% more virtual links, and hence, in higher virtual network operation and maintenance complexity.

Table 10.1: Comparison overview of VNO-Resilience and PIP-Resilience for connectivity services. The indications "+" and "-" mean that a specific model brings benefits or drawbacks in terms of the corresponding metric, respectively. "•" means that a general conclusion is not possible and the results depend on the used settings.

| **Metric** | **VNO-Resilience** | **PIP-Resilience** |
|---|---|---|
| Virtual network cost | • | • |
| Service latency | + | - |
| Resource requirements | + | - |
| Virtual network complexity | - | + |
| Network resilience knowhow | - | + |

Table 10.1 provides an overview of the above mentioned findings. The lack of network resilience knowhow at a VNO can be a problem, and it might require additional costs in terms of staff and training, and therefore, might be a decisive metric in resilience design layer selection. However, due to its ownership of the network architecture, a PIP possesses this knowledge already, and hence, it is not a problem for it. Finally, in cases where the PIP and VNO business roles are realized by the same company, the knowledge and virtual network cost metrics loose their importance.

Chapter 6 first provides an initial analysis for the selection of the resilience layer for cloud connections for existing virtual networks. As before, virtual and physical layer resilience options are considered. Two scenarios are evaluated, where in the first one the same virtual network is used for both models and in the second one optimal resilient virtual design models are utilized to design the initial virtual networks without cloud services, which have different topological properties with VNO-Resilience and PIP-Resilience. Both scenarios show a clear benefit of using virtual layer resilience for cloud connections. Our quantitative study shows that the latency gain of virtual layer resilience compared with having it in the physical layer is about 60% if the same virtual network is used for both models and reaches 120% for virtual network topologies designed with the corresponding resilience options. Using the end-to-end optimization models spanning both the network and cloud domains, the maximum guaranteed latency gain of virtual layer resilience reaches 210% for the same settings.

Using the end-to-end optimization models, we differentiate four alternatives in terms of resilience, where the models provide resilience in the virtual or in the physical layer for both network and IT domains, or delegate the network resilience to the physical layer while providing the DC resilience in the virtual layer. For the last case we consider two options, namely protecting all paths, i.e. the paths leading to the primary DC site as well as leading to the DR sites, namely Hybrid All paths Protected (HAP), and protecting only the primary path, which is called the Hybrid Primary Protected (HPP) model. Comparing the performance of these four alternatives, we observe different behaviors using different cost settings as in the case with connectivity services. In terms of virtual network setup cost, VNO-Resilience outperforms the others in case of the dominance of the link cost. The amount of excess cost compared with VNO-Resilience is 50-130%, 50-80% and 140-170% for PIP-Resilience, HPP, and HAP, respectively. For other cost settings, PIP-Resilience provides a lower cost, where in case of similar link, node and VM costs, the difference is insignificant, namely around 5%. Only in case of network node cost dominance, the difference reaches 40%. The hybrid models, HAP and HPP, always result in the highest cost values, where for the cases of node or VM cost dominance their difference with VNO-Resilience results vanishes.

Table 10.2: Comparison overview of VNO-Resilience, PIP-Resilience, HAP and HPP for cloud services. The indications ”+” and ”-” mean that a specific model brings benefits or drawbacks in terms of the corresponding metric, respectively. ”++” and ”–” indicate that these benefits and drawbacks are significant, respectively. ”•” means that a general conclusion is not possible and the results depend on the used settings.

| **Metric** | **VNO-Resilience** | **PIP-Resilience** | **HAP** | **HPP** |
|---|---|---|---|---|
| Virtual network cost | • | • | - | - |
| Service latency | ++ | − | + | ++ |
| Resource requirements | ++ | − | − | + |
| Virtual network complexity | − | ++ | − | − |
| Network resilience knowhow | - | + | + | + |

In terms of latency, PIP-Resilience performs the worst, exceeding the double amount of latency observed with VNO-Resilience as mentioned before, due to limited DC choice of a PIP for the DR site selection and the routing of the services to the DR site over the primary site in the physical layer. HAP follows PIP-Resilience with around 20% increased latency compared with VNO-Resilience, and VNO-Resilience and HPP perform similar to each other and have the lowest latency results.

Similar to latency results, in terms of physical network utilization, VNO-Resilience performs best, which is followed in the ascending order of resource requirements by the HPP, PIP-Resilience and HAP, where the last two require more than double network resources compared with VNO-Resilience, and HPP causes around 40% increase compared with VNO-Resilience. The performance difference between the models increases when the virtual link cost depends on its physical length. The reason of high resource requirement of PIP-Resilience is the link-level resilience provisioning and for HAP the increased level of resilience.

Finally, in terms of virtual network complexity PIP-Resilience offers the best solution followed by HAP, VNO-Resilience and HPP, where they result in up to 100% more virtual links compared with PIP-Resilience. Using fixed cost values, the last three result in the same number of virtual links.

All in all, virtual layer resilience shows the highest benefits in terms of latency and network utilization and can offer some gain in terms of cost depending on the used cost setting. However, it results in higher virtual network complexity. Hybrid resilience models show a weak cost performance but a high latency gain similar to VNO-Resilience. In cases, where a VNO wants to delegate the network resilience to the PIP(s) due to e.g. lack of the necessary knowledge of network operation, they can provide feasible solutions. An overview of all the results for cloud services is provided in Table 10.2.

The results for connectivity and cloud services are aligned in terms of general trends. The higher impact of the resilience layer decision on cost, latency and resource requirements for cloud services is due to the increased limitations of the physical layer in routing and selecting the DC sites. For connectivity services, the main difference between the two layers arises from the granularity of the protection. VNO-Resilience can provide end-to-end path protection for the services, whereas in case of PIP-Resilience the virtual links are protected leading to a segment protection model from the perspective of the services. For the case of cloud services, there is additionally the limitation due to the routing structure and DR site selection as mentioned above.

The aforementioned chapters deal with resilience in case of single link and node failures, where the protection paths are dedicated to each service. We now consider two enhanced models, where shared protection is allowed, as presented in Chapter 7 and where the resilience level is increased to protect against double link or node failures, virtual layer failures and geographical failures as introduced in Chapter 9.

Regarding shared protection, sharing of the redundant resources is realized at the same layer at which resilience is provisioned. For connectivity services, if the link cost is dominant, VNO-Resilience performs better, where the cost difference is increased to 90%, but the network resource requirement gain slightly decreases compared with the dedicated protection case to around 35%. For the cost setting with equal link and node costs and fixed link cost values, it is better to apply PIP-Resilience as in the case of dedicated protection, which results in 15% lower cost and 30% lower network resource requirement compared with VNO-Resilience. For equal link and node costs with length dependent link cost, the resource requirement gain of VNO-Resilience compared with PIP-Resilience increases slightly compared with dedicated protection and reaches 20%, where the cost behavior remains similar. For the dominance of capacity-dependent link cost a similar behavior is observed for shared and dedicated protection cases.

Using the shared protection models for cloud services, in terms of network utilization, VNO-Resilience is always better than PIP-Resilience by 50-300%. In terms of DC utilization the performance depends on the number of DCs per dcPIP since a PIP is limited within its domain for DR site selection and sharing. The cost performance varies with different cost settings. For the cost setting with dominant and length-dependent link cost, VNO-Resilience is always better by 65-200%, where the difference increases with increasing number of DCs and services. For node cost dominance, PIP-Resilience always yields around 40% lower cost due to the higher number of virtual nodes required by VNO-Resilience. For the remaining cost settings the two models perform similarly. In terms of service latency and virtual network setup complexity, the difference of VNO-Resilience and PIP-Resilience is not much affected by the usage of shared protection and is comparable to the case of dedicated protection.

In conclusion, sharing the redundant virtual resources increases the differences between the virtual and physical layer resilience alternatives under some conditions, however, does not affect the overall trends in their performance compared with dedicated protection results.

Finally, if the protection level is increased, the resilience provisioning layer decision depends more on the type of failures against which the network is to be protected. For both models, having protection against geographical or sub-network failures does not increase the virtual network cost compared with protection against single physical link or node failures. Cost overhead for protection against double link failures is up to 46% for PIP-Resilience and 34% for VNO-Resilience, and against virtual node failures is up to 26% for VNO-Resilience for the used cost setting in Chapter 9. However, certain failures can be only recovered from at the virtual layer. If protection against virtual layer failures (virtual link and node failures) is already in place in the VNO-Resilience model, protection against physical layer failures (physical link and node failures) is implicitly provided without any additional cost, and hence, virtual layer resilience should be preferred.

# 11. Conclusion and Outlook

This thesis solves the end-to-end resilient virtual network design problem both for connectivity and cloud services. The solutions are modeled as Mixed-Integer Linear Programmings (MILPs) and heuristic algorithms. The models are then enhanced for extended failure coverage and for enabling shared protection and Quality of Service (QoS) provisioning in virtual networks. Finally, via extensive simulations, the performance of the proposed models and algorithms is evaluated, which build a framework for the decision on the resilience and QoS provisioning layer for the operators in their future network design.

### Resilient Virtual Network Design

Designing efficient end-to-end resilient solutions especially for cloud services is mainly impossible today due to the separate operation of network and cloud domains. Moreover, the cloud traffic is often sent over the Internet as a best-effort traffic, and hence, end-to-end performance guarantees cannot be provided either. Virtual networks offer a solution to this problem due to the scope they enable over heterogeneous domains. However, how to design such an end-to-end resilient and optimized network is up to now an open question as the existing literature cannot provide a complete answer to it. The literature from overlay networks lacks the optimization of the virtual network mapping, and from Virtual Private Networks (VPNs) and virtual network embedding service routing optimization, respectively. Therefore, they can only provide sub-optimal solutions. In this thesis, we enable simultaneous optimization for service routing and virtual network mapping - in other words, having a cost-optimal resilience virtual network design supporting an end-to-end high performance, which is crucial for a Virtual Network Operator (VNO) and enables a cost-efficient design for future networks, which supports the realization of new technologies like Network Functions Virtualization (NFV).

We model the virtual network architecture with the interactions of the business roles. Based on this model, we formulate MILPs to design virtual networks, and propose corresponding heuristics. We introduce novel virtual network design models for connectivity and cloud services, where the latter considers both the options of connecting an existing virtual network to the cloud domain and designing an end-to-end optimized virtual network for cloud services spanning network and cloud infrastructures. Due to the scalability problems of the MILP solutions under certain circumstances, we propose the HillClimber and kBest heuristics for virtual network design, which are evaluated to be scalable and perform close to optimal.

### Enhanced Design Models for Virtual Networks

The aforementioned models protect virtual networks against single physical link or node failures. However, there are various types of failures, which can occur in a virtual network environment. Some failures like virtual link and Virtual Machine (VM) failures are only recoverable from at the virtual layer due to separation of the operation of the different layers. Moreover, a Physical Infrastructure Provider (PIP) also has difficulties to recover from Data Center (DC) and sub-network failures due to its domain-based limitations in recovery. However, it has the advantage of being close to the origin of the failure in case of physical failures.

Analyzing the trade-off between the resilience level and its cost to an operator, our results show that both VNO-Resilience and PIP-Resilience models can offer protection against sub-network failures without causing any increase in virtual network cost compared with protection against single physical link and node failures. Protection against double link failures or VM failures comes with a higher cost ranging between around 30-50%. Therefore, the decision for the protection level for these cases should be taken according to the Service Level Agreement (SLA) requirements and the cost benefit this additional protection level offers in terms of network operation. Finally, it is advised to use VNO-Resilience to protect against physical link/node failures if protection against virtual link and node failures is already in place as the latter includes the former, and hence, does not cause any price increase of the virtual network.

Resilience is a crucial part of network design but has a high cost to both business roles. To lower the cost of dedicated protection, in today's networks shared protection mechanisms are used, which allow the sharing of redundant resources between services, whose primary paths do not share selected set of common risks. We apply this concept to virtual networks, where we allow the sharing of redundant virtual resources. Shared protection in virtual networks creates a win-win situation for the VNOs and the PIPs. It lowers the virtual resource usage, and hence, decreases the cost of the virtual network for a VNO. At the same time, it improves the utilization of the physical resources for a PIP, allowing it to serve more customers with its available resources.

To enable shared protection in virtual networks, in this thesis we introduce both the necessary architecture extensions and the required enhancements to the virtual network design models. The architectural extension enables the sharing of redundant virtual resources by exchanging a certain level of knowledge between the VNO and the PIP without the disclosure of business-critical information. This information exchange is, however, necessary since otherwise the PIP lacks the service routing knowledge and the VNO the information about topological mapping of the virtual resources, which prevents them from being able to apply shared protection separately.

We use both MILPs and heuristics for the design of virtual networks with shared protection. Applied in the virtual layer, shared protection lowers the cost by up to 30% and the resource requirements by up to 70% for connectivity services. With PIP-Resilience no cost reduction can be observed due to independence of the cost and the sharing of the resources in the physical layer, however, the amount of required resources can be reduced by up to 50% for the same virtual network cost. For cloud services, also redundant DC resources can be shared among different services, and shared protection brings 10-20% improvement in cost and resource requirements both on the network and on the cloud side.

QoS provisioning is another essential part of network design, which is especially important for business-critical or latency sensitive services. In a virtual network environment, there are two interfaces, where QoS provisioning plays an important role, namely between a VNO and a Service Provider (SP) in terms of guaranteeing a certain quality for the connectivity or cloud services, and between a PIP and a VNO for the QoS guarantees of the virtual resources. The former denotes e.g. the bandwidth, computational resource and end-to-end latency requirements of a service. The latter guarantees to a VNO that e.g. a virtual link has a certain maximum latency as it belongs to a certain QoS class. For this purpose, we introduce the concept of QoS classes to the virtual networks, where they are applied both to define the service requirements and the properties of virtual resources. We conduct simulations to observe the necessity of having QoS provisioning in virtual networks. If network design does not consider service requirements, our results show a remarkably deteriorated service delivery quality, where around 70% of the services are affected by this problem. This points out that investing to support QoS guarantees is a concrete need, and its absence can be very costly for businesses.

We differentiate between three alternatives, where both QoS and resilience provisioning are in the physical layer, and where QoS is provisioned in the virtual layer with resilience provisioning being in the virtual or physical layer. Our results show that for connectivity services deploying QoS at the VNO layer while delegating resilience provisioning to the PIP layer outperforms significantly the other two models in terms of the virtual network setup cost. Nevertheless, considering PIP-QoS with PIP-Resilience results in a better network resource utilization. Supporting both service differentiation and resilience at the VNO layer results in the middle of the two other models in terms of cost and network utilization, and hence, it can serve as a trade-off solution. For cloud services, however, PIP-Resilience has the additional routing restriction that the services need to be re-routed from the primary site to the protection site instead of re-routing at the service source node. This and the Disaster Recovery (DR) site selection restriction are the main reasons for VNO-QoS with VNO-Resilience results being better than the physical layer solutions both in terms of the virtual network setup cost and network utilization.

### Decision on Resilience Provisioning Layer

In the virtual network architecture, there are three fundamental ways of providing resilience. It can be provisioned in the virtual layer by the VNO, in the physical layer by the PIP(s) or a combination of these two approaches can be used. PIP-Resilience is similar to the traditional resilience mechanisms in terms of having recovery in the physical layer. Allowing resilience provisioning in the virtual layer has certain advantages and drawbacks. In terms of resource utilization, the physical layer enjoys having complete information about its resources, however, it is restricted within its own domain in providing resilience. In the virtual layer, resilience design can be performed using an overview of the advertised resources of different physical domains but a VNO does not necessarily have a full knowledge of the PIP domain. Therefore, both of these options lack an overall optimization. In terms of service level resilience, the virtual layer has more benefits since it possesses the knowledge about the services, whereas in the physical layer the design is done with the virtual resource granularity. In terms of complexity, physical layer resilience is more advantageous due to scalability and signaling issues. However, since a VNO has an overview of different physical domains, it has an advantage in terms of service latency. In terms of cost, there is a trade-off between acquiring a fewer amount of resilient and more expensive virtual resources and a higher amount of cheaper non-resilient resources. All in all, the decision of resilience layer is dependent of different metrics. Therefore, the performance of different layer resilience solutions need to be further analyzed quantitatively.

This evaluation is performed for all the introduced models and algorithms in terms of virtual network setup cost, service latency, resource requirements and complexity. For these evaluations, we use simulations, which are conducted using the Java Virtual Network Simulator that has been developed in the framework of this thesis. Our simulation results are in a ±5% confidence interval with a 95% confidence level. They show that virtual layer resilience provides the highest benefits in terms of latency and network utilization reaching more than 50%, and it can offer a certain gain in terms of cost depending on the used cost setting. However, it results in higher virtual network complexity compared with physical layer resilience. Hybrid resilience models, which use a combination of the two approaches, have a weak cost performance but a high latency gain compared with PIP-Resilience similar to VNO-Resilience due to their virtual layer routing properties. Therefore, they can be a good compromise for the cases, where a VNO prefers to delegate network resilience to the physical layer due to e.g. lack of the necessary knowledge of network operation.

The simulation results are aligned for connectivity and cloud services. However, the differences of the two resilience options is more emphasized in case of cloud services, which is due to the higher amount of limitations a PIP faces when routing cloud services. In the case of connectivity services, the performance difference of the two approaches in terms of cost, latency and resource utilization arises from the granularity of the protection as mentioned above. For the case of cloud services, there is additionally the restriction in selecting the protection site DC from the same cloud provider domain and DR site routing over the primary site in case of physical layer resilience, whereas a VNO possesses the freedom to select the DC sites from various provider domains and to route the services directly from the service source nodes to the primary and DR sites.

### Future Work

The virtual network design problem is a very important problem for future networks to which this thesis is offering solutions with the consideration of different resilience and QoS concepts. It is a new and broad domain, for which a lot of further research is required in order to increase the efficiency, flexibility and reliability of future networks. In the following, we list some examples of future research directions.

This thesis focuses on the research questions about resilience in virtual networks for the design phase of these networks. This is an initial and important step in this area. With time, during the operation of the virtual networks, however, there can be certain changes in the services using the virtual network. Therefore, a future step would be to investigate the methods for virtual networks to adapt to such changes. This can be achieved for example by adding new virtual links, nodes or VMs to the network or by adjusting their capacities accordingly. Moreover, a re-routing of the existing services might be desired for re-optimization, which however can cause disruptions of these services. This trade-off is already an important topic in the literature for today's networks and should be also considered for the case of virtual networks. Similar to the addition of new services, the tear down of the services should be also analyzed, which can be addressed by removal or down-sizing of virtual resources and re-routing of remaining services if necessary.

Moreover, in our models we mainly incorporate protection strategies, which are essential for high service quality and reliability provisioning. A further idea would be using a mixture of protection and restoration mechanisms, especially for enhanced failure coverage, where against the first failure protection mechanisms can serve, and further recovery can be offered by restoration means. Finally, another aspect for DC resilience is the synchronization of the primary and protection sites, which can be incorporated into the proposed virtual network design models by modeling the synchronization traffic and including the synchronization path to the optimization model.

# Part VI

# Appendix

# A. Glossary

## A.1 Simulation Parameters

This section aims to summarize all the simulation parameters used in different evaluation throughout the thesis. It serves as a quick reference for each chapter, listing all the relevant simulation parameter values. Providing this information enables the reproduction of the conducted experiments. The parameter values of each chapter and section will be given separately in different subsections, where the title of the subsection refers to the corresponding model in the thesis.

### A.1.1 Optimization Models for Resilient Virtual Network Design

In this subsection, the parameter setting for the results evaluation in Section 5.4 are provided as given A.1. The same parameters are used in the performance evaluation of the shared protection models for connectivity services, which are presented in Section 7.3.

| **Parameter** | **Value** |
|---|---|
| Used physical topology | NobelEU, NobelUS |
| Service type | Unicast |
| Number of service nodes | 3-10 |
| Demand matrix | Request service between each service node pair |
| Bandwidth request per service | 1 unit |
| Node request per service | 1 unit |
| Resilience premium | 2 |

Table A.1: Parameter setting for the evaluation in Section 5.4.

### A.1.2 Cloud Extension Models for Random Virtual Networks

In this subsection, the parameter setting for the results evaluation in Section 6.2.1 are provided as given A.2.

### A.1.3 Cloud Extension Models for Resilient Virtual Network Design

In this subsection, the parameter setting for the results evaluation in Section 6.2.2 are provided as given A.3.

| Parameter | Value |
|---|---|
| Used physical topology | Random topologies generated according to [87] with 30, 60 and 100 physical nodes |
| Service type | Anycast (Cloud) |
| Number of availability regions | 6, 12 and 20 |
| Number of service nodes | 3-10 |
| Number of dcPIPs | 1-30 |
| Number of DCs per dcPIP | 2-4 |
| Demand matrix | Request service from each service node to an arbitrary DC |
| Bandwidth request per service | 1 unit |
| Node request per service | 1 unit |
| VM request per service | 1 unit |
| Location of DCs | Random, Farthest |
| PIP-Resilience strategy | Shortest Delay, Random Selection |

Table A.2: Parameter setting for the evaluation in Section 6.2.1.

### A.1.4 Combined Optimization Models for Resilient Virtual Network Design with Cloud Services

In this subsection, the parameter setting for the results evaluation in Section 6.4.2 are provided as given A.4. The same parameters are used in the performance evaluation of the shared protection models for cloud services, which are presented in Section 7.4.

### A.1.5 Optimization Models for Resilient and QoS-Aware Virtual Network Design

In this subsection, the parameter settings for the results evaluation in Section 8.2.4 are provided as given A.5.

### A.1.6 Optimization Models for Resilient and QoS-Aware Virtual Network Design for Cloud Services

In this subsection, the parameter settings for the results evaluation in Section 8.3.5 are provided as given A.6.

### A.1.7 Optimization Models for Resilient Virtual Network Design with Extended Failure Coverage

In this subsection, the parameter setting for the results evaluation in Chapter 9 are provided as given A.7.

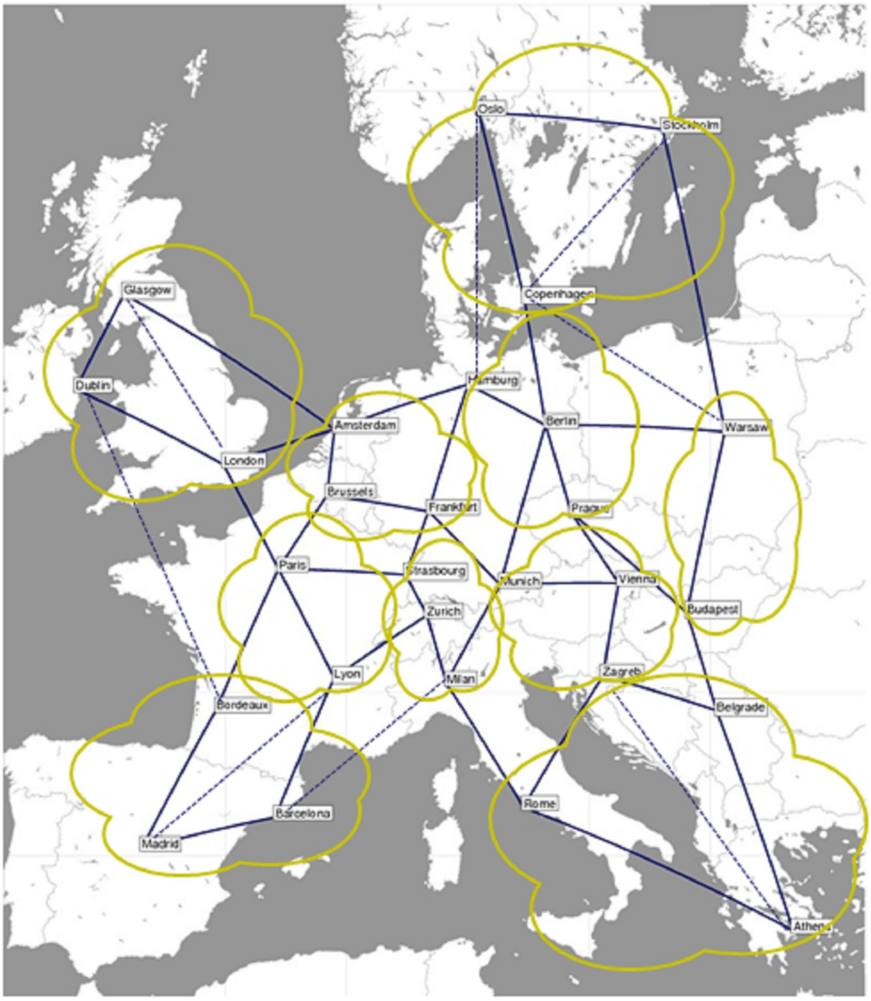


Figure A.1: Modified NobelEU network topology for the use of extended failure coverage with added connectivity denoted as dashed lines and subnetwork regions marked with cloud shapes [205]

| **Parameter** | **Value** |
|---|---|
| Used physical topology | NobelUS and NobelEU topologies |
| Service type | Anycast (Cloud) |
| Number of availability regions | 6 and 12 |
| Number of service nodes | 3-10 |
| Number of dcPIPs | 1-10 |
| Number of DCs per dcPIP | 2 |
| Demand matrix | Request service from each service node to an arbitrary DC |
| Bandwidth request per service | 1 unit |
| Node request per service | 1 unit |
| VM request per service | 1 unit |
| Location of DCs | Random, Farthest |
| PIP-Resilience strategy | Random Selection |

Table A.3: Parameter setting for the evaluation in Section 6.2.2.

| **Parameter** | **Value** |
|---|---|
| Used physical topology | NobelEU and NobelUS topologies |
| Service type | Anycast (Cloud) |
| Number of service nodes | 1-10 |
| Number of dcPIPs | 1-6 |
| Number of DCs per dcPIP | 2-10 |
| Demand matrix | Request service from each service node to an arbitrary DC |
| Bandwidth request per service | 1 unit |
| Node request per service | 1 unit |
| VM request per service | 1 unit |
| Resilience premium | 2 |
| Location of DCs | Random, Farthest |
| PIP-Resilience strategy | Shortest Delay, Random Selection |

Table A.4: Parameter setting for the evaluation in Section 6.4.2.

| **Parameter** | **Value** |
|---|---|
| Used physical topology | NobelEU |
| Service type | Unicast |
| Number of service nodes | 3-6 |
| Demand matrix | Request service between each service node pair |
| Bandwidth request per service | 10 units |
| Capacity of a physical link | 240 units |
| Node request per service | 1 unit |
| Resilience premium | 2 |
| Fixed link cost (Gold/Silver/Bronze) | 500/300/200 units |
| Capacity dependent link cost (Gold/Silver/Bronze) | 10/6/4 units |
| Fixed node cost (Gold/Silver/Bronze) | 50/30/20 units |
| Capacity dependent node cost (Gold/Silver/Bronze) | 10/6/4 units |
| Service delay requirements for Gold, Silver and Bronze service class | 20, 70 and 170 ms |
| Service distributions (Gold/Silver/Bronze) | 40/30/30%, 60/20/20% and 20/60/20% |

Table A.5: Parameter setting for the evaluation in Section 8.2.4.

| **Parameter** | **Value** |
|---|---|
| Used physical topology | NobelEU |
| Service type | Anycast (Cloud) |
| Number of service nodes | 3-5 |
| Number of dcPIPs | 1 |
| Number of DCs per dcPIP | 4 |
| DCs locations | London, Warsaw, Madrid and Rome |
| Demand matrix | Request service from each service node to an arbitrary DC |
| Bandwidth request per service | 1 unit |
| Node request per service | 1 unit |
| VM request per service | 1 unit |
| Resilience premium | 2 |
| PIP-Resilience strategy | Random Selection |
| Fixed link cost (Gold/Silver/Bronze) | 500/300/200 units |
| Capacity dependent link cost (Gold/Silver/Bronze) | 10/6/4 units * hopcount of the physical path |
| Fixed node cost (Gold/Silver/Bronze) | 50/30/20 units |
| Capacity dependent node cost (Gold/Silver/Bronze) | 10/6/4 units |
| Fixed VM cost (Gold/Silver/Bronze) | 100/60/30 units |
| Capacity dependent VM cost (Gold/Silver/Bronze) | 10/6/4 units |
| Service delay requirements for Gold, Silver and Bronze service class | 40, 70 and 100 ms |
| Thresholds for link class assignment ($< x$ gold, $< y$ silver, $> y$ bronze) | $x = 15\%$, $y = 25\%$ |
| Service distributions (Gold/Silver/Bronze) | 40/30/30% for both delay and VM requirements |

Table A.6: Parameter setting for the evaluation in Section 8.3.5.

| **Parameter** | **Value** |
|---|---|
| Used physical topology | NobelEU modified as to be 3-node-connected as shown in Figure A.1 |
| Service type | Unicast |
| Number of service nodes | 3-6 |
| Demand matrix | Request service between each service node pair |
| Bandwidth request per service | 1 unit |
| Node request per service | 1 unit |
| Resilience premium | 2 |

Table A.7: Parameter setting for the evaluation in Chapter 9.

# List of Acronyms

| | |
|---|---|
| **AS** | Autonomous System |
| **ASBR** | Autonomous System Border Router |
| **BGP** | Border Gateway Protocol |
| **CPU** | Central Processing Unit |
| **DC** | Data Center |
| **dcPIP** | Data Center PIP |
| **DR** | Disaster Recovery |
| **EBGP** | External Border Gateway Protocol |
| **E-NNI** | External Network Network Interface |
| **ERO** | Explicit Route Object |
| **ETSI** | European Telecommunications Standards Institute |
| **JUNG** | Java Universal Network/Graph Framework |
| **GML** | Graph Modeling Language |
| **GMPLS** | Generalized Multi-Protocol Label Switching |
| **GRE** | Generic Routing Encapsulation |
| **GW** | Gateway |
| **HAP** | Hybrid All paths Protected |
| **HPP** | Hybrid Primary Protected |
| **IP** | Internet Protocol |
| **IPLS** | IP-only LAN-like Service |
| **IPsec** | Internet Protocol Security |
| **ISP** | Internet Service Provider |
| **LAN** | Local Area Network |
| **LP** | Linear Programming |
| **LSP** | Label Switched Path |
| **MILP** | Mixed-Integer Linear Programming |

| | |
|---|---|
| **MPLS** | Multi-Protocol Label Switching |
| **nPIP** | Network PIP |
| **NFV** | Network Functions Virtualization |
| **OF** | OpenFlow |
| **OSPF** | Open Shortest Path First |
| **PIP** | Physical Infrastructure Provider |
| **QoE** | Quality of Experience |
| **QoS** | Quality of Service |
| **RRO** | Record Route Object |
| **RSVP-TE** | Resource reSerVation Protocol Traffic Engineering |
| **SDN** | Software Defined Networking |
| **SERO** | Secondary Explicit Route Object |
| **SLA** | Service Level Agreement |
| **SP** | Service Provider |
| **SPM** | Shortest Path Mapping |
| **SPMwAN** | Shortest Path Mapping with Additional Nodes |
| **SRLG** | Shared Risk Link Group |
| **SRRO** | Secondary Record Route Object |
| **UNI** | User Network Interface |
| **VM** | Virtual Machine |
| **VNO** | Virtual Network Operator |
| **VNP** | Virtual Network Provider |
| **VPLS** | Virtual Private LAN Service |
| **VPN** | Virtual Private Network |
| **VPWS** | Virtual Private Wire Service |
| **VRF** | Virtual Routing and Forwarding |
| **VXLAN** | Virtual Extensible Local Area Network |
| **WAN** | Wide Area Network |
| **WDM** | Wavelength-Division Multiplexing |

# List of Symbols

$a_G$ Average shortest disjoint path pair length in a physical topology $T$

$a_p$ Average length of all the paths in shortest disjoint path pairs in a physical network

$a_s$ Average length of the disjoint paths calculated over all possible nodes of the physical network for the paths leading from that node to all possible node pairs

$a_{s,c}$ Binary variable taking the value of 1 if a virtual machine is placed into the DC connected to node $c \in C$ to satisfy the anycast demand with source $s \in S$, 0 otherwise

$a_{s,c,b}$ Binary variable taking the value of 1 if a Bronze class virtual machine is placed into the DC connected to node $c \in C_G$ to satisfy the anycast demand with source s $\in$ S, 0 otherwise

$a_{s,c,g}$ Binary variable taking the value of 1 if a Gold class virtual machine is placed into the DC connected to node $c \in C_G$ to satisfy the anycast demand with source s $\in$ S, 0 otherwise

$a_{s,c,s}$ Binary variable taking the value of 1 if a Silver class virtual machine is placed into the DC connected to node $c \in C_G$ to satisfy the anycast demand with source s $\in$ S, 0 otherwise

$a'_{s,c}$ Binary variable taking the value of 1 if a virtual machine is placed into the DC connected to node $c \in C$ to be used as the DR site of the anycast demand with source $s \in S$, 0 otherwise

$b_d$ Requested bandwidth for the service $d \in D$

$B_l$ Set of the physical links $e \in E$, on which the virtual protection (backup) link $l \in L$ is mapped

$C$ Set of the DC connection nodes

$C_B$ set of DC connection nodes containing Bronze class VM

$C_G$ set of DC connection nodes containing Gold class VM

$C_l$ Bandwidth request of a new service on virtual link $l$

$C_S$ set of DC connection nodes containing Silver class VM

$C_t$ Total amount of requested bandwidth in the virtual network

$C_{t,l}$ Total amount of requested bandwidth on virtual link $l$

$C_{t,v}$ Total amount of requested node resources on virtual node $v$

$C_v$ Node resource request of a new service on virtual node $v$

$D$ Set of the requested services

| | |
|---|---|
| $d_c$ | Delay on the path pair connecting primary DC node $c$ to protection DC site |
| $d_{c,b}$ | Delay on Bronze path pair connecting node $c$ to protection DC node |
| $d_{c,g}$ | Delay on Gold path pair connecting node $c$ to protection DC node |
| $d_{c,s}$ | Delay on Silver path pair connecting node $c$ to protection DC node |
| $d_d$ | Requested end-to-end delay for service $d \in D$ |
| $d_l$ | End-to-end delay on a virtual link $l \in L$ |
| $D_u$ | Set of the all possible unicast realizations of the requested cloud services, where $\lvert D_u \rvert = \lvert S \rvert \cdot \lvert C \rvert$ and $d = (s, c) \in D_u$ with $s \in S$ and $c \in C$ |
| $D_s$ | Set of the all possible unicast realizations of the requested cloud service having the source node $s \in S$ with $\lvert D_s \rvert = \lvert C \rvert$ and $D_s \subseteq D_u$ |
| $E$ | Set of the edges in the physical network topology |
| $e_{cap}$ | Available capacity on a physical edge $e \in E$ |
| $e_G$ | Edge length of a square area within which a physical network topology $G$ is generated according to [87] |
| $E_l$ | Set of the endpoints of link $l \in L$ |
| $E_l^r$ | Set of the endpoints of link $l \in L_r$ |
| $E_l^n$ | Set of the endpoints of link $l \in L_n$ |
| $G$ | Physical network topology |
| $G_l$ | Virtual network topology |
| $g_{s,c,c'}$ | Binary variable taking the value of 1 if the server $c' \in C$ is used for the protection of the server $c \in C$ for the anycast service with source node $s \in S$, 0 otherwise |
| $L$ | Set of the all virtual link candidates |
| $L_b$ | Set of virtual links that are of Bronze class |
| $L_g$ | Set of virtual links that are of Gold class |
| $L_r$ | Set of resilient virtual links |
| $L_n$ | Set of virtual links mapped on single physical paths (no resilience provisioning) |
| $l_s$ | Length of the path $s$ |
| $L_s$ | Set of virtual links that are of Silver class |
| $m_p$ | Average of the shortest disjoint path pair lengths between each physical node pair |
| $m_s$ | Average of the minimum path lengths for each physical node to two arbitrary DC locations |
| $N$ | Set of the nodes in the physical network topology |
| $n_c$ | Used protection capacity on server $c \in C$ |
| $n_{c,c'}$ | Used capacity on server $c' \in C$ that is used for the protection of the server $c \in C$ |
| $n_d$ | Requested node resources for the service $d \in D$ |
| $n_{dc}$ | Number of the DCs, which will be selected for each cloud service with $n_{dc} \in \{1, ..., \lvert C \rvert\}$ |

| | |
|---|---|
| $n_l$ | Number of virtual links in the virtual network |
| $n_v$ | Number of virtual nodes in the virtual network |
| $n_s$ | Number of the services in the virtual network |
| $P_l$ | Set of the physical edges $e \in E$, on which the virtual link $l \in L$ is mapped |
| $p_{n,m}$ | Shortest disjoint path pair between the nodes $n$ and $m$ |
| $q_{dc}$ | Number of all available DCs connected to the physical network |
| $q_{dc_i}$ | Number of the DCs belonging to a dcPIP $i$ |
| $R$ | Set of DC connection node pairs $(c_1, c_2) \in C^2$ with $c_1 \neq c_2$, which are located in the same availability region of the physical topology |
| $r_d$ | Requested server resources for the service $d \in D_u$ |
| $r_{DC}$ | Cost factor of the DC resource utilization in the cost objective function |
| $r_{NU}$ | Cost factor of the network utilization in the cost objective function |
| $r_{\text{PIP}}$ | Resilience premium for providing resilience for a certain virtual network element in the physical layer |
| $r_{\text{VNO}}$ | Additional cost factor for providing virtual node protection at the VNO |
| $S$ | Set of the service nodes |
| $S_B$ | Set of services classified as Bronze (requires Bronze, Silver or Gold VM) |
| $S_G$ | Set of services classified as Gold (requires Gold VM) |
| $S_S$ | Set of services classified as Silver (requires Silver or Gold VM) |
| $s_e$ | Length of physical edge $e \in E$ |
| $s_{n_1,n_2}$ | Shortest disjoint path from the node $n_1$ to $n_2$ |
| $t_l$ | Physical length of link $l \in L$ |
| $u_l$ | Used capacity on link $l \in L$ |
| $V$ | Set of the all virtual node candidates |
| $v_c$ | Used working capacity on server $c \in C$ |
| $v_{cap}$ | Available resources on a virtual node $v \in V$, which is the available resources of the physical node on which $v$ is mapped |
| $W_l$ | Set of the physical links $e \in E$, on which the virtual working link $l \in L$ is mapped |
| $y_c$ | Binary variable taking the value of 1 if a virtual machine on the DC connected to node $c \in C$ is included to the virtual network, 0 otherwise |
| $y_{c,b}$ | Binary variable taking the value of 1 if a Bronze class virtual machine on the DC connected to node $c \in C$ is in the resulting virtual network, 0 otherwise |
| $y_{c,g}$ | Binary variable taking the value of 1 if a Gold class virtual machine on the DC connected to node $c \in C$ is in the resulting virtual network, 0 otherwise |
| $y_{c,s}$ | Binary variable taking the value of 1 if a Silver class virtual machine on the DC connected to node $c \in C$ is in the resulting virtual network, 0 otherwise |
| $y_e$ | Used capacity on physical edge $e \in E$, $y_e \in [0, \infty]$ |
| $Z$ | Set of virtual links $(j, k) \in L^2$ , which share at least one physical edge/node |

$Z'$ Set of virtual links $(j,k) \in L^2$ , which share at least one physical node from the same region except for the source and destination nodes i.e. not region disjoint

$z_c$ Used capacity on DC connected to node $c \in C$ with $z_c \in [0, \infty)$

$z_{c,b}$ Used capacity on a Bronze class DC connected to node $c \in C_B$

$z_{c,g}$ Used capacity on a Gold class DC connected to node $c \in C_G$

$z_{c,s}$ Used capacity on a Silver class DC connected to node $c \in C_S$

$\alpha_{b,v}$ Binary variable taking the value of 1 if a node $v \in V$ is of Bronze class in the resulting virtual network, 0 otherwise

$\alpha_{g,v}$ Binary variable taking the value of 1 if a node $v \in V$ is of Gold class in the resulting virtual network, 0 otherwise

$\alpha_{gs,v}$ Binary variable taking the value of 1 if a node $v \in V$ is of Gold or Silver class in the resulting virtual network, 0 otherwise

$\alpha_{s,v}$ Binary variable taking the value of 1 if a node $v \in V$ is of Silver class in the resulting virtual network, 0 otherwise

$\alpha_v$ Binary variable taking the value of 1 if the node $v \in V$ is in the resulting virtual network, 0 otherwise

$\beta_{d,l}$ Binary variable taking the value of 1 if the link $l \in L$ is used for the demand $d \in D_u$ and if demand $d = (s,c)$ is chosen as one of the realizations of the cloud service with source $s \in S$, 0 otherwise

$\beta_{i,d,l}$ Binary variable taking the value of 1 if the link $l \in L$ is used for the $i^{th}$ route of the demand $d \in D$, 0 otherwise

$\gamma_l$ Binary variable taking the value of 1 if the link $l \in L$ is in the resulting virtual network, 0 otherwise

$\delta_{b,d,v}$ Binary variable taking the value of 1 if a node $v \in V$ is used for the route of the cloud service realization $d \in D$ and its selected class is Bronze, 0 otherwise

$\delta_{b,i,d,v}$ Binary variable taking the value of 1 if a node $v \in V$ is used for the $i^{th}$ route of the demand $d \in D$ and its selected class is Bronze, 0 otherwise

$\delta_{d,v}$ Binary variable taking the value of 1 if the node $v \in V$ is used for the demand $d \in D_u$ and if demand $d = (s,c)$ is chosen as one of the realizations of the cloud service with source $s \in S$, 0 otherwise

$\delta_{i,d,v}$ Binary variable taking the value of 1 if the node $v \in V$ is used for the $i^{th}$ route of the demand $d \in D$, 0 otherwise

$\delta_{g,d,v}$ Binary variable taking the value of 1 if a node $v \in V$ is used for the route of the cloud service realization $d \in D$ and its selected class is Gold, 0 otherwise

$\delta_{g,i,d,v}$ Binary variable taking the value of 1 if a node $v \in V$ is used for the $i^{th}$ route of the demand $d \in D$ and its selected class is Gold, 0 otherwise

$\delta_{s,d,v}$ Binary variable taking the value of 1 if a node $v \in V$ is used for the route of the cloud service realization $d \in D$ and its selected class is Silver, 0 otherwise

$\delta_{s,i,d,v}$ Binary variable taking the value of 1 if a node $v \in V$ is used for the $i^{th}$ route of the demand $d \in D$ and its selected class is Silver, 0 otherwise

$\varepsilon$ Setup cost of virtual network

$\epsilon_{j,d,e}$ Binary variable taking the value of 1 if the physical edge $e \in E$ is used for the $j^{th}$ route of the demand $d \in D$ on the physical substrate, 0 otherwise

$\eta_v$ Setup cost per unit capacity for node $v \in V$

$\eta_{b,v}$ Setup cost per unit resource for a bronze node $v \in V$

$\eta_{g,v}$ Setup cost per unit resource for a gold node $v \in V$

$\eta_{s,v}$ Setup cost per unit resource for a silver node $v \in V$

$\theta_l$ Setup cost per unit capacity for link $l \in L$

$\kappa_c$ Used capacity on server $c \in C$ that is used for the protection of the servers calculated according to the working path disjointness criterion

$\kappa_{c,l}$ Used capacity on server $c \in C$ that is used for the protection of the servers with working paths containing $l \in L$

$\Lambda$ Set of DCs with $\Lambda_p \in \Lambda$ and $\Lambda_b \in \Lambda$ denote a primary or DR site if selected as such, respectively

$\lambda_l$ Fixed setup cost for having a new link $l \in L$

$\mu_v$ Fixed setup cost for having a new node $v \in V$

$\mu_{b,v}$ Fixed setup cost for having a bronze node $v \in V$

$\mu_{g,v}$ Fixed setup cost for having a gold node $v \in V$

$\mu_{s,v}$ Fixed setup cost for having a silver node $v \in V$

$\nu_{d,e,e'}$ Binary variable taking the value of 1 if the physical edge $e' \in E$ is used for the protection of the physical edge $e \in E$ for the demand $d \in D$ on the physical substrate, 0 otherwise

$\xi_{d,l,c}$ Binary variable taking the value 1 if a service $d = (s, c) \in D_s$ used the DC $c$ as its backup DC and the virtual link $l \in L$ is used as part of the service's working path, 0 otherwise

$\pi_e$ Used protection capacity on physical edge $e \in E$, $\pi_e \in [0, \infty]$

$\pi_{e,e'}$ Used capacity on physical edge $e' \in E$ that is used for the protection of the physical edge $e \in E$, $\pi_{e,e'} \in [0, \infty]$

$\rho_e$ Used working capacity on physical edge $e \in E$, $\rho_e \in [0, \infty]$

$\varrho_g$ Setup cost per unit capacity for Gold path pair to protection DC site

$\varrho_s$ Setup cost per unit capacity for Silver path pair to protection DC site

$\varrho_b$ Setup cost per unit capacity for Bronze path pair to protection DC site

$\Sigma$ Set of the regions $\Xi \in \Sigma$, where each region $\Xi$ is again a set of the DCs which are in that region

$\tau_{d,c,l}$ Binary variable taking the value of 1 if the link $l \in L$ is part of the protection path of $d \in D_s$ and the server $c \in C$ is used as the primary site of $d$, 0 otherwise

$\tau_{d_1,d_2,l,l'}$ Binary variable taking the value of 1 if the link $l' \in L$ is used for the protection of the link $l \in L$, where $l'$ is part of the path of $d_2 \in D_s$ and it is a backup path, i.e. $i \geq 2$, and $l$ is part of the path of $d_1 \in D_s$ and it is the primary path, 0 otherwise

$\tau_{d,l,l'}$ Binary variable taking the value of 1 if the link $l' \in L$ is used for the protection of the link $l \in L$ for the demand $d \in D$, 0 otherwise

$\phi_c$ Fixed setup cost for having a new virtual machine in the virtual network, which is connected to node $c \in C$

| | |
|---|---|
| $\phi_{b,c}$ | Fixed setup cost for having a Bronze VM connected to node $c \in C$ |
| $\phi_{g,c}$ | Fixed setup cost for having a Gold VM connected to node $c \in C$ |
| $\phi_{s,c}$ | Fixed setup cost for having a Silver VM connected to node $c \in C$ |
| $\varphi_c$ | Setup cost per unit capacity of a virtual machine connected to node $c \in C$ |
| $\varphi_{b,c}$ | Setup cost per unit capacity for a Bronze VM connected to node $c \in C$ |
| $\varphi_{g,c}$ | Setup cost per unit capacity for a Gold VM connected to node $c \in C$ |
| $\varphi_{s,c}$ | Setup cost per unit capacity for a Silver VM connected to node $c \in C$ |
| $\phi_{c,l}$ | Used capacity on link $l \in L$ that is used for the protection of the server $c \in C$ |
| $\phi_l$ | Used protection capacity on link $l \in L$, $\phi_l \in [0, \infty]$ |
| $\phi_{l,l'}$ | Used capacity on link $l' \in L$, that is used for the protection of the link $l \in L$, $\phi_{l,l'} \in [0, \infty]$ |
| $\chi_{d,b}$ | Binary variable taking the value of 1 if a Bronze class path pair connecting primary DC node to protection DC is selected by the service realization $d \in D$ in PIP-Resilience, 0 otherwise |
| $\chi_{d,g}$ | Binary variable taking the value of 1 if a Gold class path pair connecting primary DC node to protection DC is selected by the service realization $d \in D$ in PIP-Resilience, 0 otherwise |
| $\chi_{d,s}$ | Binary variable taking the value of 1 if a Silver class path pair connecting primary DC node to protection DC is selected by the service realization $d \in D$ in PIP-Resilience, 0 otherwise |
| $\psi_l$ | Used working capacity on link $l \in L$, $\psi_l \in [0, \infty]$ |
| $\omega_{b,v}$ | Used capacity on a Bronze node $v \in V$ |
| $\omega_{g,v}$ | Used capacity on a Gold node $v \in V$ |
| $\omega_{s,v}$ | Used capacity on a Silver node $v \in V$ |
| $\omega_v$ | Used capacity on node $v \in V$ |

# Literature

[1] Albert Greenberg, James Hamilton, David A. Maltz, and Parveen Patel. The cost of a cloud: Research problems in data center networks. *SIGCOMM Comput. Commun. Rev.*, 39(1):68–73, December 2008.

[2] M. Pioro and D. Medhi. *Routing, Flow, and Capacity Design in Communication and Computer Networks*. Elsevier, 2004.

[3] Symantec. Virtualization and evolution to the cloud survey, 2011.

[4] Ponemon Institute. 2013 cost of data center outages, 2013.

[5] C. Strachey. Time sharing in large fast computers. In *Proceedings of the International Conference on Information processing, UNESCO*, 1959.

[6] Aameek Singh, Madhukar Korupolu, and Dushmanta Mohapatra. Server-storage virtualization: Integration and load balancing in data centers. In *Proceedings of the 2008 ACM/IEEE Conference on Supercomputing*, SC '08, pages 53:1–53:12, Piscataway, NJ, USA, 2008. IEEE Press.

[7] Nelson Ruest and Danielle Ruest. *Virtualization, A Beginner's Guide*. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 2009.

[8] N.M. Mosharaf Kabir Chowdhury and Raouf Boutaba. A survey of network virtualization. *Computer Networks*, 54(5):862 – 876, 2010.

[9] H.A. Seid and A.L. Lespagnol. Virtual private network, 1998. US Patent 5,768,271.

[10] E. Rosen and Y. Rekhter. BGP/MPLS VPNs. RFC 2547 (Informational), March 1999. Obsoleted by RFC 4364.

[11] E. Rosen and Y. Rekhter. BGP/MPLS IP Virtual Private Networks (VPNs). RFC 4364 (Proposed Standard), February 2006. Updated by RFCs 4577, 4684, 5462.

[12] L. Andersson and T. Madsen. Provider Provisioned Virtual Private Network (VPN) Terminology. RFC 4026 (Informational), March 2005.

[13] M. Carugi and D. McDysan. Service Requirements for Layer 3 Provider Provisioned Virtual Private Networks (PPVPNs). RFC 4031 (Informational), April 2005.

[14] R. Callon and M. Suzuki. A Framework for Layer 3 Provider-Provisioned Virtual Private Networks (PPVPNs). RFC 4110 (Informational), July 2005.

[15] D. Farinacci, T. Li, S. Hanks, D. Meyer, and P. Traina. Generic Routing Encapsulation (GRE). RFC 2784 (Proposed Standard), March 2000. Updated by RFC 2890.

[16] G. Dommety. Key and Sequence Number Extensions to GRE. RFC 2890 (Proposed Standard), September 2000.

[17] C. Perkins. IP Encapsulation within IP. RFC 2003 (Proposed Standard), October 1996. Updated by RFCs 3168, 6864.

[18] A. Conta and S. Deering. Generic Packet Tunneling in IPv6 Specification. RFC 2473 (Proposed Standard), December 1998.

[19] S. Kent and R. Atkinson. Security Architecture for the Internet Protocol. RFC 2401 (Proposed Standard), November 1998. Obsoleted by RFC 4301, updated by RFC 3168.

[20] S. Kent and R. Atkinson. IP Authentication Header. RFC 2402 (Proposed Standard), November 1998. Obsoleted by RFCs 4302, 4305.

[21] E. Rosen, A. Viswanathan, and R. Callon. Multiprotocol Label Switching Architecture. RFC 3031 (Proposed Standard), January 2001. Updated by RFCs 6178, 6790.

[22] B. Davie, J. Lawrence, K. McCloghrie, E. Rosen, G. Swallow, Y. Rekhter, and P. Doolan. MPLS using LDP and ATM VC Switching. RFC 3035 (Proposed Standard), January 2001.

[23] W. Augustyn and Y. Serbest. Service Requirements for Layer 2 Provider-Provisioned Virtual Private Networks. RFC 4665 (Informational), September 2006.

[24] L. Andersson and E. Rosen. Framework for Layer 2 Virtual Private Networks (L2VPNs). RFC 4664 (Informational), September 2006.

[25] T. Takeda. Framework and Requirements for Layer 1 Virtual Private Networks. RFC 4847 (Informational), April 2007.

[26] John Jannotti, David K. Gifford, Kirk L. Johnson, M. Frans Kaashoek, and James W. O'Toole, Jr. Overcast: Reliable multicasting with on overlay network. In *Proceedings of the 4th Conference on Symposium on Operating System Design & Implementation - Volume 4*, OSDI'00, pages 14–14, Berkeley, CA, USA, 2000. USENIX Association.

[27] T. Anderson, L. Peterson, S. Shenker, and J. Turner. Overcoming the internet impasse through virtualization. *Computer*, 38(4):34–41, April 2005.

[28] PlanetLab: An open platform for developing, deploying, and accessing planetary-scale services. http://www.planet-lab.org/.

[29] James P.G. Sterbenz, Deep Medhi, Byrav Ramamurthy, Caterina Scoglio, David Hutchison, Bernhard Plattner, Tricha Anjali, Andrew Scott, Cort Buffington, Gregory E. Monaco, Don Gruenbacher, Rick McMullen, Justin P. Rohrer, John Sherrell, Pragatheeswaran Angu, Ramkumar Cherukuri, Haiyang Qian, and Nidhi Tare. The great plains environment for network innovation (GpENI): A programmable testbed for future internet architecture research. In *Proceedings of the 6th International Conference on Testbeds and Research Infrastructures for the Development of Networks & Communities (TridentCom)*, pages 428–441, Berlin, Germany, May 18–20 2010.

[30] Ramkumar Cherukuri, Xuan Liu, Andy Bavier, James P.G. Sterbenz, and Deep Medhi. Network virtualization in gpeni: Framework, implementation and integration experience. *The 3rd IEEE/IFIP International Workshop on Management of the Future Internet (ManFI)*, May 2011.

[31] GENI - Global Environment for Network Innovations. http://www.GENI.net.

[32] Onelab. https://onelab.eu/.

[33] L. Peterson, S. Sevinc, J. Lepreau, R. Ricci, J. Wroclawski, S. Faber, T. amd Schwab, and S. Baker. Slice-based facility architecture, 2009.

[34] T. Aoyama. A new generation network: Beyond the internet and ngn. *Communications Magazine, IEEE*, 47(5):82–87, May 2009.

[35] Nick Feamster, Lixin Gao, and Jennifer Rexford. How to lease the internet in your spare time. *SIGCOMM Comput. Commun. Rev.*, 37(1):61–64, January 2007.

[36] Gregor Schaffrath, Christoph Werle, Panagiotis Papadimitriou, Anja Feldmann, Roland Bless, Adam Greenhalgh, Andreas Wundsam, Mario Kind, Olaf Maennel, and Laurent Mathy. Network virtualization architecture: Proposal and initial prototype. In *Proceedings of the 1st ACM Workshop on Virtualized Infrastructure Systems and Architectures*, VISA '09, pages 63–72, New York, NY, USA, 2009. ACM.

[37] Jorge Carapinha and Javier Jiménez. Network virtualization: A view from the bottom. In *Proceedings of the 1st ACM Workshop on Virtualized Infrastructure Systems and Architectures*, VISA '09, pages 73–80, New York, NY, USA, 2009. ACM.

[38] M. Hoffmann and M. Staufer. Network virtualization for future mobile networks: General architecture and applications. In *Communications Workshops (ICC), 2011 IEEE International Conference on*, pages 1–5, June 2011.

[39] E. Escalona, Shuping Peng, R. Nejabati, D. Simeonidou, J.A Garcia-Espin, J. Ferrer, S. Figuerola, G. Landi, N. Ciulli, J. Jimenez, B. Belter, Y. Demechenko, C. De Laat, Xiaomin Chen, A Yukan, S. Soudan, P. Vicat-Blanc, J. Buysse, M. De Leenheer, C. Develder, A. Tzanakaki, P. Robinson, M. Brogle, and T.M. Bohnert. Geysers: A novel architecture for virtualization and co-provisioning of dynamic optical networks and it services. In *Future Network Mobile Summit (FutureNetw), 2011*, pages 1–8, June 2011.

[40] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. Openflow: Enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev.*, 38(2):69–74, March 2008.

[41] Nick Feamster, Jennifer Rexford, and Ellen Zegura. The road to sdn. *Queue*, 11(12):20:20–20:40, December 2013.

[42] Hyojoon Kim and N. Feamster. Improving network management with software defined networking. *Communications Magazine, IEEE*, 51(2):114–119, February 2013.

[43] NSN white paper, Technology Vision for the Gigabit Experience, Technology Vision 2020, June 2013.

[44] Ericsson white paper, The Real-Time Cloud, February 2014.

[45] Alcatel-Lucent white paper, Charting the course to a virtualized carrier network, 2013.

[46] Huawei white paper, 5G: A Technology Vision, 2013.

[47] Network function virtualization, white paper. In *SDN and OpenFlow World Congress, Darmstadt, Germany*, 2012.

[48] D. Medhi and K. Ramasamy. *Network routing: algorithms, protocols, and architectures.* Morgan Kaufmann, 2010.

[49] A. Leon-Garcia and I. Widjaja. *Communication networks: fundamental concepts and key architectures.* McGraw-Hill Professional, 2004.

[50] R. Ramaswami and K.N. Sivarajan. *Optical networks: a practical perspective.* Morgan Kaufmann, 2002.

[51] A. Markopoulou, Gianluca Iannaccone, Supratik Bhattacharyya, Chen-Nee Chuah, Y. Ganjali, and C. Diot. Characterization of failures in an operational ip backbone network. *Networking, IEEE/ACM Transactions on*, 16(4):749–762, Aug 2008.

[52] A.K. Somani. *Survivability and traffic grooming in WDM optical networks.* Cambridge University Press, 2006.

[53] M. Haider. The impact of network downtime on businesses today, market insight paper, networks first limited, 2010.

[54] I.B. Barla. Network Coding for Transport Network Resilience, Master's thesis, Technische Universität München, 2009.

[55] M. Ilyas and H.T. Mouftah. *The handbook of optical communication networks.* CRC Press, 2003.

[56] C. Ou and B. Mukherjee. *Survivable optical WDM networks.* Springer, 2005.

[57] Evolven, Downtime, Outages and Failures - Understanding Their True Costs. http://www.evolven.com/blog/downtime-outages-and-failures-understanding-their-true-costs.html.

[58] A. Arnold. Assessing the financial impact of downtime. http://www.businesscomputingworld.co.uk/assessing-the-financial-impact-of-downtime/, 2010.

[59] C. Harris. It downtime costs $26.5 billion in lost revenue. http://www.informationweek.com/it-downtime-costs-\protect\T1\textdollar265-billion-in-lost-revenue, 2011.

[60] C. Kachris, K. Kanonakis, and I Tomkos. Optical interconnection networks in data centers: recent trends and future challenges. *Communications Magazine, IEEE*, 51(9):39–45, September 2013.

[61] M. Alicherry and T. V. Lakshman. Network aware resource allocation in distributed clouds. In *INFOCOM, 2012 Proceedings IEEE*, pages 963–971, March 2012.

[62] Hui Yang, Jie Zhang, Yongli Zhao, Hui Li, Shanguo Huang, Yuefeng Ji, Jianrui Han, Yi Lin, and Young Lee. Cross stratum resilience for openflow-enabled data center interconnection with flexi-grid optical networks. *Optical Switching and Networking*, 11, Part A(0):72 – 82, 2014.

[63] B. Kantarci and H.T. Mouftah. Minimum outage probability provisioning in an energy-efficient cloud backbone. In *Global Communications Conference (GLOBECOM), 2013 IEEE*, pages 2879–2884, Dec 2013.

[64] K. Schmidt. *High Availability and Disaster Recovery: Concepts, Design, Implementation.* Springer, 2006.

[65] G.B. Dantzig. *Linear Programming and Extensions.* Princeton University Press, 1963.

[66] A.M. Verweij. *Selected Applications of Integer Programming: A Computational Study*. PhD thesis, Universiteit Utrecht, 2000.

[67] ILOG CPLEX 12.3, http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/.

[68] Gurobi optimization, mixed integer programming basics. http://www.gurobi.com/resources/getting-started/mip-basics.

[69] Zbigniew Michalewicz and David B. Fogel. *How to Solve It: Modern Heuristics*. Springer, 2004.

[70] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2 edition, 2003.

[71] Kwan wai Wong, Chi-Ying Tsui, R.S.-K. Cheng, and Wai-Ho Mow. A vlsi architecture of a k-best lattice decoding algorithm for mimo channels. In *Circuits and Systems, 2002. ISCAS 2002. IEEE International Symposium on*, volume 3, pages III–273–III–276 vol.3, 2002.

[72] I.B. Barla Harter, D.A. Schupke, M. Hoffmann, and G. Carle. Network virtualization for disaster resilience of cloud services. In *IEEE Communications Magazine*. (Accepted).

[73] I.B. Barla, D.A. Schupke, and G. Carle. Analysis of resilience in virtual networks. In *11th Würzburg Workshop on IP: Joint ITG and Euro-NF Workshop Visions of Future Generation Networks*, 2011.

[74] L. Fang, D. Fernando, R. andRao, and S. Boutros. BGP/MPLS IP VPN data center interconnect, Internet draft (work in progress), IETF, Oct. 2013, draft-fang-l3vpn-data-center-interconnect-02, 2013.

[75] Sebastian Meier, Marc Barisch, Andreas Kirstädter, Daniel Schlosser, Michael Duelli, Michael Jarschel, Tobias Hoßfeld, Klaus Hoffmann, Marco Hoffmann, Wolfgang Kellerer, et al. Provisioning and operation of virtual networks. *Electronic Communications of the EASST*, 37, 2011.

[76] N.M.M.K. Chowdhury and R. Boutaba. Network virtualization: state of the art and research challenges. *Communications Magazine, IEEE*, 47(7):20–26, July 2009.

[77] Qiang Duan, Yuhong Yan, and AV. Vasilakos. A survey on service-oriented network virtualization toward convergence of networking and cloud computing. *Network and Service Management, IEEE Transactions on*, 9(4):373–392, December 2012.

[78] Fang Hao, T. V. Lakshman, Sarit Mukherjee, and Haoyu Song. Enhancing dynamic cloud-based services using network virtualization. In *Proceedings of the 1st ACM Workshop on Virtualized Infrastructure Systems and Architectures*, VISA '09, pages 37–44, New York, NY, USA, 2009. ACM.

[79] Guohui Wang and T.S.E. Ng. The impact of virtualization on network performance of amazon ec2 data center. In *INFOCOM, 2010 Proceedings IEEE*, pages 1–9, March 2010.

[80] K.R. Jackson, L. Ramakrishnan, K. Muriki, S. Canon, S. Cholia, J. Shalf, Harvey J. Wasserman, and N.J. Wright. Performance analysis of high performance computing applications on the amazon web services cloud. In *Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on*, pages 159–168, Nov 2010.

[81] The year in downtime top 10 outages of 2012 and 2013. http://www.datacenterknowledge.com, 2013.

[82] Yi Wang, Eric Keller, Brian Biskeborn, Jacobus Merwe, and Jennifer Rexford. Virtual routers on the move: Live router migration as a network-management primitive. In *SIGCOMM CCR*, 2008.

[83] I.B. Barla, D.A. Schupke, and G. Carle. Delay performance of resilient cloud services over networks. In *Parallel and Distributed Processing with Applications (ISPA), 2012 IEEE 10th International Symposium on*, pages 512–517, July 2012.

[84] Java Universal Network/Graph Framework, http://jung.sourceforge.net/.

[85] ILOG Concert Technology, http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/interfaces/.

[86] B.M. Waxman. Routing of multipoint connections. *Selected Areas in Communications, IEEE Journal on*, 6(9):1617–1622, Dec 1988.

[87] Claunir Pavan, Rui Manuel Morais, José R. Ferreira da Rocha, and Armando Nolasco Pinto. Generating realistic optical transport network topologies. *J. Opt. Commun. Netw.*, 2(1):80–90, Jan 2010.

[88] Isil Burcu Barla, Dominic A. Schupke, and Georg Carle. Resilient virtual network design for end-to-end cloud services. In *Proceedings of the 11th International IFIP TC 6 Conference on Networking - Volume Part I*, IFIP'12, pages 161–174, Berlin, Heidelberg, 2012. Springer-Verlag.

[89] David Andersen, Hari Balakrishnan, Frans Kaashoek, and Robert Morris. *Resilient overlay networks*, volume 35. ACM, 2001.

[90] Z. Li and P. Mohapatra. Qron: Qos-aware routing in overlay networks. *Selected Areas in Communications, IEEE Journal on*, 22(1):29–40, Jan 2004.

[91] Mina Kamel, Caterina Scoglio, and Todd Easton. Optimal topology design for overlay networks. In IanF. Akyildiz, Raghupathy Sivakumar, Eylem Ekici, Jaudelice-Cavalcantede Oliveira, and Janise McNair, editors, *NETWORKING 2007. Ad Hoc and Sensor Networks, Wireless Networks, Next Generation Internet*, volume 4479 of *Lecture Notes in Computer Science*, pages 714–725. Springer Berlin Heidelberg, 2007.

[92] Markosz Maliosz and Tibor Cinkler. Configuration of protected virtual private networks. In *Design Of Reliable Communication Networks, DRCN 2001*, 2001.

[93] Yong Zhu and M. Ammar. Algorithms for assigning substrate network resources to virtual network components. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1–12, April 2006.

[94] N.M.M.K. Chowdhury, M.R. Rahman, and R. Boutaba. Virtual network embedding with coordinated node and link mapping. In *INFOCOM 2009, IEEE*, pages 783–791, April 2009.

[95] E. Modiano and A. Narula-Tam. Survivable lightpath routing: a new approach to the design of wdm-based networks. *Selected Areas in Communications, IEEE Journal on*, 20(4):800–809, May 2002.

[96] M.R. Rahman, I. Aib, and R. Boutaba. Survivable virtual network embedding. In Mark Crovella, LauraMarie Feeney, Dan Rubenstein, and S.V. Raghavan, editors, *NETWORKING 2010*, volume 6091 of *Lecture Notes in Computer Science*, pages 40–52. Springer Berlin Heidelberg, 2010.

[97] P. Demeester, M. Gryseels, A. Autenrieth, C. Brianza, L. Castagna, G. Signorelli, R. Clemenfe, M. Ravera, A. Jajszczyk, D. Janukowicz, K. Van Doorselaere, and Y. Harada. Resilience in multilayer networks. *Communications Magazine, IEEE*, 37(8):70–76, Aug 1999.

[98] F. Touvet and D. Harle. Network resilience in multilayer networks: A critical review and open issues. In Pascal Lorenz, editor, *Networking - ICN 2001*, volume 2093 of *Lecture Notes in Computer Science*, pages 829–837. Springer Berlin Heidelberg, 2001.

[99] D. Medhi and D. Tipper. Multi-layered network survivability-models, analysis, architecture, framework and implementation: an overview. In *DARPA Information Survivability Conference and Exposition, 2000. DISCEX '00. Proceedings*, volume 1, pages 173–186 vol.1, 2000.

[100] Nobel US and Nobel EU topologies. http://sndlib.zib.de.

[101] D.A. Schupke. Multilayer and multidomain resilience in optical networks. *Proceedings of the IEEE*, 100(5):1140–1148, May 2012.

[102] L. Massoulie, A.-M. Kermarrec, and A.J. Ganesh. Network awareness and failure resilience in self-organizing overlay networks. In *Reliable Distributed Systems, 2003. Proceedings. 22nd International Symposium on*, pages 47–55, Oct 2003.

[103] J.P.G. Sterbenz, E.K. Cetinkaya, M.A. Hameed, A. Jabbar, and J.P. Rohrer. Modelling and analysis of network resilience. In *Communication Systems and Networks (COMSNETS), 2011 Third International Conference on*, pages 1–10, Jan 2011.

[104] EgemenK. Cetinkaya, Dan Broyles, Amit Dandekar, Sripriya Srinivasan, and JamesP.G. Sterbenz. Modelling communication network challenges for future internet resilience, survivability, and disruption tolerance: a simulation-based approach. *Telecommunication Systems*, 52(2):751–766, 2013.

[105] A. Autenrieth and A. Kirstadter. Engineering end-to-end ip resilience using resilience-differentiated qos. *Communications Magazine, IEEE*, 40(1):50–57, Jan 2002.

[106] A. Basta, I.B. Barla, M. Hoffmann, and G. Carle. Qos-aware optimal resilient virtual networks. In *Communications (ICC), 2013 IEEE International Conference on*, pages 2251–2255, June 2013.

[107] R. Diestel. *Graph Theory*. Springer, 2000.

[108] Shengli Yuan and Jason P Jue. Heuristic routing algorithm for shared protection in connection-oriented networks. In *OptiComm 2001: Optical Networking and Communications Conference*, pages 142–152. International Society for Optics and Photonics, 2001.

[109] Rajiv Ramaswami and Kumar N Sivarajan. Design of logical topologies for wavelength-routed optical networks. *Selected Areas in Communications, IEEE Journal on*, 14(5):840–851, 1996.

[110] Dhritiman Banerjee and Biswanath Mukherjee. Wavelength-routed optical networks: Linear formulation, resource budgeting tradeoffs, and a reconfiguration study. *IEEE/ACM Transactions on Networking (TON)*, 8(5):598–607, 2000.

[111] Tibor Cinkler, Dániel Marx, Claus Popp Larsen, and Dániel Fogaras. Heuristic algorithms for joint configuration of the optical and electrical layer in multi-hop wavelength routing networks. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 1000–1009. IEEE, 2000.

[112] Vachaspathi P Kompella, Joseph C Pasquale, and George C Polyzos. Multicast routing for multimedia communication. *IEEE/ACM Transactions on Networking (TON)*, 1(3):286–292, 1993.

[113] Xin Yuan and Xingming Liu. Heuristic algorithms for multi-constrained quality of service routing. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 844–853. IEEE, 2001.

[114] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms.* The MIT Press, 2009.

[115] Shigang Chen and Klara Nahrsted. An overview of quality of service routing for next-generation high-speed networks: problems and solutions. *Network, IEEE*, 12(6):64–79, 1998.

[116] Rudra Dutta and George N Rouskas. A survey of virtual topology design algorithms for wavelength routed optical networks. *Optical Networks Magazine*, 1(1):73–89, 2000.

[117] C. Gruber. *Design and Optimization of Resilient Multipath Networks.* PhD thesis, Technische Universität München, 2007.

[118] S. Ramamurthy, L. Sahasrabuddhe, and B. Mukherjee. Survivable wdm mesh networks. *Lightwave Technology, Journal of*, 21(4):870–883, April 2003.

[119] K. Walkowiak and J. Rak. Shared backup path protection for anycast and unicast flows using the node-link notation. In *Communications (ICC), 2011 IEEE International Conference on*, pages 1–6, June 2011.

[120] Panagiotis Papadimitriou, Olaf Maennel, Adam Greenhalgh, Anja Feldmann, and Laurent Mathy. Implementing network virtualization for a future internet. In *20th ITC Specialist Seminar on Network Virtualization-Concept and Performance Aspects*, 2009.

[121] P. Vicat-Blanc, S. Figuerola, X. Chen, G. Landi, E. Escalona, C. Develder, A. Tzanakaki, Y. Demchenko, J. Garcia Espin, J. Ferrer, E. Lopez, S. Soudan, J. Buysse, A. Jukan, N. Ciulli, M. Brogle, L. van Laarhoven, B. Belter, F. Anhalt, R. Nejabati, D. Simeonidou, C. Ngo, C. de Laat, M. Biancani, M. Roth, P. Donadio, J. Jimenez, M. Antoniak-Lewandowska, and A. Gumaste. Bringing optical networks to the cloud: An architecture for a sustainable future internet. In John Domingue, Alex Galis, Anastasius Gavras, Theodore Zahariadis, Dave Lambert, Frances Cleary, Petros Daras, Srdjan Krco, Henning Müller, Man-Sze Li, Hans Schaffers, Volkmar Lotz, Federico Alvarez, Burkhard Stiller, Stamatis Karnouskos, Susanna Avessta, and Michael Nilsson, editors, *The Future Internet*, volume 6656 of *Lecture Notes in Computer Science*, pages 307–320. Springer Berlin Heidelberg, 2011.

[122] J. Van der Merwe, K.K. Ramakrishnan, M. Fairchild, A. Flavel, J. Houle, H.A. Lagar-Cavilla, and J. Mulligan. Towards a ubiquitous cloud computing infrastructure. In *Local and Metropolitan Area Networks (LANMAN), 2010 17th IEEE Workshop on*, pages 1–6, May 2010.

[123] Amazon AWS. http://aws.amazon.com/directconnect/, 2013.

[124] I.B. Barla, D.A. Schupke, M. Hoffmann, and G. Carle. Optimal design of virtual networks for resilient cloud services. In *Design of Reliable Communication Networks (DRCN), 2013 9th International Conference on the*, pages 218–225, March 2013.

[125] I.B. Barla Harter, D.A. Schupke, M. Hoffmann, and G. Carle. Optimal design of resilient virtual networks. In *Journal of Optical Communications and Networking (JOCN), Invited article.* (Accepted).

[126] S. Koo, G. Sahin, and S.S. Subramaniam. Cost efficient lsp protection in ip/mpls-over-wdm overlay networks. In *Communications, 2003. ICC '03. IEEE International Conference on*, volume 2, pages 1278–1282 vol.2, May 2003.

[127] A. Capone, Jocelyne Elias, and F. Martignon. Models and algorithms for the design of service overlay networks. *Network and Service Management, IEEE Transactions on*, 5(3):143–156, September 2008.

[128] Kayi Lee, Eytan Modiano, and Hyang-Won Lee. Cross-layer survivability in wdm-based networks. *IEEE/ACM Trans. Netw.*, 19(4):1000–1013, August 2011.

[129] Hongfang Yu, Chunming Qiao, Vishal Anand, Xin Liu, Hao Di, and Gang Sun. Survivable virtual infrastructure mapping in a federated computing and networking system under single regional failures. In *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pages 1–6, Dec 2010.

[130] D. Dietrich, A. Rizk, and P. Papadimitriou. Multi-domain virtual network embedding with limited information disclosure. In *IFIP Networking Conference, 2013*, pages 1–9, May 2013.

[131] Wai-Leong Yeow, Cedric Westphal, and Ulas C. Kozat. Designing and embedding reliable virtual infrastructures. *SIGCOMM Comput. Commun. Rev.*, 41(2):57–64, April 2011.

[132] J.W. Jiang, Tian Lan, Sangtae Ha, Minghua Chen, and Mung Chiang. Joint vm placement and routing for data center traffic engineering. In *INFOCOM, 2012 Proceedings IEEE*, pages 2876–2880, March 2012.

[133] Jielong Xu, Jian Tang, K. Kwiat, Weiyi Zhang, and Guoliang Xue. Survivable virtual infrastructure mapping in virtualized data centers. In *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*, pages 196–203, June 2012.

[134] M. Alicherry and T. V. Lakshman. Network aware resource allocation in distributed clouds. In *INFOCOM, 2012 Proceedings IEEE*, pages 963–971, March 2012.

[135] M.A. Owens and D. Medhi. Temporal bandwidth-intensive virtual network allocation optimization in a data center network. In *Communications (ICC), 2013 IEEE International Conference on*, pages 3493–3497, June 2013.

[136] T. L. Weems. How far is far enough. *Disaster Recovery J.*, 16(2), Spring 2003.

[137] Canhui Ou, Jing Zhang, Hui Zang, L.H. Sahasrabuddhe, and B. Mukherjee. New and improved approaches for shared-path protection in wdm mesh networks. *Lightwave Technology, Journal of*, 22(5):1223–1232, May 2004.

[138] Shengli Yuan, Jason P Jue, et al. Shared protection routing algorithm for optical network. *Optical Networks Magazine*, 3(3):32–39, 2002.

[139] I.B. Barla, K. Hoffmann, M. Hoffmann, D.A. Schupke, and G. Carle. Shared protection in virtual networks. In *IEEE ICC 2013 Workshop on Clouds Networks and Data Centers*, June 2013.

[140] I.B. Barla Harter, M. Hoffmann, D.A. Schupke, and G. Carle. Scalable resilient virtual network design algorithms for cloud services. In *6th International Workshop on Reliable Networks Design and Modeling (RNDM)*, 2014 (Accepted).

[141] Pin-Han Ho and H.T. Mouftah. A framework for service-guaranteed shared protection in wdm mesh networks. *Communications Magazine, IEEE*, 40(2):97–103, Feb 2002.

[142] C. Ou, Keyao Zhu, Hui Zang, L.H. Sahasrabuddhe, and B. Mukherjee. Traffic grooming for survivable wdm networks - shared protection. *Selected Areas in Communications, IEEE Journal on*, 21(9):1367–1383, Nov 2003.

[143] Pin-Han Ho, J. Tapolcai, H.T. Mouftah, and Chi-Hsiang Yeh. Linear formulation for path shared protection. In *Communications, 2004 IEEE International Conference on*, volume 3, pages 1622–1627, June 2004.

[144] Pin-Han Ho, J. Tapolcai, and H.T. Mouftah. On achieving optimal survivable routing for shared protection in survivable next-generation internet. *Reliability, IEEE Transactions on*, 53(2):216–225, June 2004.

[145] Pin-Han Ho and H.T. Mouftah. Shared protection in mesh wdm networks. *Communications Magazine, IEEE*, 42(1):70–76, Jan 2004.

[146] Pin-Han Ho, J. Tapolcai, and T. Cinkler. Segment shared protection in mesh communications networks with bandwidth guaranteed tunnels. *Networking, IEEE/ACM Transactions on*, 12(6):1105–1118, Dec 2004.

[147] Dahai Xu, Yizhi Xiong, and Chunming Qiao. Novel algorithms for shared segment protection. *Selected Areas in Communications, IEEE Journal on*, 21(8):1320–1331, Oct 2003.

[148] Ali Shaikh, Jens Buysse, Brigitte Jaumard, and Chris Develder. Anycast routing for survivable optical grids: scalable solution methods and the impact of relocation. *JOURNAL OF OPTICAL COMMUNICATIONS AND NETWORKING*, 3(9):767–779, 2011.

[149] C.G. Gruber, A M C A Koster, S. Orlowski, R. Wessaly, and A Zymolka. A computational study for demand-wise shared protection. In *Design of Reliable Communication Networks, 2005. (DRCN 2005). Proceedings.5th International Workshop on*, pages 8 pp.–, Oct 2005.

[150] Arie M. C. A. Koster, Christoph Gerlach, and Christoph Gerlacht. Demand-wise shared protection for meshed optical networks. *Journal of Network and Systems Management*, 13:35–55, 2005.

[151] Hongfang Yu, Vishal Anand, Chunming Qiao, and Gang Sun. Cost efficient design of survivable virtual infrastructure to recover from facility node failures. In *Communications (ICC), 2011 IEEE International Conference on*, pages 1–6, June 2011.

[152] W. Yeow, C. Westphal, and U. Kozat. Designing and embedding reliable virtual infrastructures. In *ACM SIGCOMM Workshop on Virtualized Infrastructure Systems and Architectures (VISA'10)*, 2010.

[153] R. Bless and C. Werle. Network virtualization from a signaling perspective. In *Communications Workshops, 2009. ICC Workshops 2009. IEEE International Conference on*, pages 1–6, June 2009.

[154] R. Aggarwal, D. Papadimitriou, and S. Yasukawa. Extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label Switched Paths (LSPs). RFC 4875 (Proposed Standard), May 2007. Updated by RFC 6510.

[155] J.P. Lang, Y. Rekhter, and D. Papadimitriou. RSVP-TE Extensions in Support of End-to-End Generalized Multi-Protocol Label Switching (GMPLS) Recovery. RFC 4872 (Proposed Standard), May 2007. Updated by RFCs 4873, 6780.

[156] A. Farrel, J.-P. Vasseur, and A. Ayyangar. A Framework for Inter-Domain Multiprotocol Label Switching Traffic Engineering. RFC 4726 (Informational), November 2006.

[157] K. Kompella and Y. Rekhter. Label Switched Paths (LSP) Hierarchy with Generalized Multi-Protocol Label Switching (GMPLS) Traffic Engineering (TE). RFC 4206 (Proposed Standard), October 2005. Updated by RFCs 6001, 6107.

[158] A. Nakao. Network virtualization as foundation for enabling new network architectures and applications. In *IEICE Transactions on Communications*, volume E93-B, pages 454 – 457. 2010.

[159] Qiang Duan. Analysis on quality of service provisioning for communication services in network virtualization. *Journal of Communications*, 7(2), 2012.

[160] Gianluca Iannaccone, Chen-nee Chuah, Richard Mortier, Supratik Bhattacharyya, and Christophe Diot. Analysis of link failures in an ip backbone. In *Proceedings of the 2Nd ACM SIGCOMM Workshop on Internet Measurment*, IMW '02, pages 237–242, New York, NY, USA, 2002. ACM.

[161] M. Bui, B. Jaumard, I.B. Barla Harter, and C. Develder. Scalable algorithms for qos-aware virtual network mapping for cloud services. In *The 18th International Conference on Optical Networking Design and Modeling (ONDM)*, 2014.

[162] Xipeng Xiao and L.M. Ni. Internet QoS: a big picture. *Network, IEEE*, 13(2):8–18, Mar 1999.

[163] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An Architecture for Differentiated Services. RFC 2475 (Informational), December 1998. Updated by RFC 3260.

[164] Shigang Chen and K. Nahrsted. An overview of quality of service routing for next-generation high-speed networks: problems and solutions. *Network, IEEE*, 12(6):64–79, Nov 1998.

[165] George Apostolopoulos, Roch Guérin, Sanjay Kamat, and Satish K. Tripathi. Quality of service based routing: A performance perspective. *SIGCOMM Comput. Commun. Rev.*, 28(4):17–28, October 1998.

[166] Z. Wang and J. Crowcroft. Quality-of-service routing for supporting multimedia applications. *Selected Areas in Communications, IEEE Journal on*, 14(7):1228–1234, Sep 1996.

[167] D. Medhi. Qos routing computation with path caching: a framework and network performance. *Communications Magazine, IEEE*, 40(12):106–113, Dec 2002.

[168] M. Boucadair, P. Levis, D. Griffin, N. Wang, M. Howarth, G. Pavlou, E. Mykoniati, P. Georgatsos, B. Quoitin, J. Rodriguez Sanchez, and M.L. Garcia-Osma. A framework for end-to-end service differentiation: Network planes and parallel internets. *Communications Magazine, IEEE*, 45(9):134–143, September 2007.

[169] Shigang Chen and K. Nahrstedt. Distributed quality-of-service routing in ad hoc networks. *Selected Areas in Communications, IEEE Journal on*, 17(8):1488–1505, Aug 1999.

[170] Hannan Xiao, W.K.-G. Seah, A Lo, and Kee Chaing Chua. A flexible quality of service model for mobile ad-hoc networks. In *Vehicular Technology Conference Proceedings, 2000. VTC 2000-Spring Tokyo. 2000 IEEE 51st*, volume 1, pages 445–449 vol.1, 2000.

[171] T. Braun, M. Guenter, and I Khalil. Management of quality of service enabled vpns. *Communications Magazine, IEEE*, 39(5):90–98, May 2001.

[172] S. Rosch and G. Algie. Dynamic virtual private network (vpn) tunnel quality of service (qos) treatment, 2000. US Patent US20050088977 A1.

[173] Haeryong Lee, Jeongyeon Hwang, Byungryong Kang, and Kyoungpyo Jun. End-to-end qos architecture for vpns: Mpls vpn deployment in a backbone network. In *Parallel Processing, 2000. Proceedings. 2000 International Workshops on*, pages 479–483, 2000.

[174] Claire Fahy, Steven Davy, Zohra Boudjemil, Sven van der Meer, JavierRubio Loyola, Joan Serrat, John Strassner, Andreas Berl, Hermann de Meer, and Daniel Macedo. Towards an information model that supports service-aware, self-managing virtual resources. In Sven van der Meer, Mark Burgess, and Spyros Denazis, editors, *Modelling Autonomic Communications Environments*, volume 5276 of *Lecture Notes in Computer Science*, pages 102–107. Springer Berlin Heidelberg, 2008.

[175] Qiang Duan. Analysis on quality of service provisioning for communication services in network virtualization. *JCM*, 7(2):143–154, 2012.

[176] Tao Guo, Ning Wang, K. Moessner, and R. Tafazolli. Shared backup network provision for virtual network embedding. In *Communications (ICC), 2011 IEEE International Conference on*, pages 1–5, June 2011.

[177] Dushyant Arora, Marcin Bienkowski, Anja Feldmann, Gregor Schaffrath, and Stefan Schmid. Online strategies for intra and inter provider service migration in virtual networks. In *Proceedings of the 5th International Conference on Principles, Systems and Applications of IP Telecommunications*, IPTcomm '11, pages 10:1–10:11, New York, NY, USA, 2011. ACM.

[178] S. Misra, S. Das, M. Khatua, and M.S. Obaidat. Qos-guaranteed bandwidth shifting and redistribution in mobile cloud environment. *Cloud Computing, IEEE Transactions on*, 2(2):181–193, April 2014.

[179] S. Shenker, C. Partridge, and R. Guerin. Specification of Guaranteed Quality of Service. RFC 2212 (Proposed Standard), September 1997.

[180] G. Rosenbaum, Chun Tung Chou, Sanjay Jha, and D. Medhi. Dynamic routing of restorable qos connections in mpls networks. In *Local Computer Networks, 2005. 30th Anniversary. The IEEE Conference on*, pages 9 pp.–426, Nov 2005.

[181] NSN white paper, lte-capable transport: A quality user experience demands an end-to-end approach, 2011.

[182] Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness.* W. H. Freeman & Co., New York, NY, USA, 1990.

[183] Z. Wang and J. Crowcroft. Quality-of-service routing for supporting multimedia applications. *Selected Areas in Communications, IEEE Journal on*, 14(7):1228–1234, Sep 1996.

[184] Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, STOC '71, pages 151–158, New York, NY, USA, 1971. ACM.

[185] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network flows: theory, algorithms, and applications.* Prentice Hall, 1993.

[186] DOCOMO tsunami report. http://www.globaltelecomsbusiness.com/Article/3054564/, 2011.

[187] A. Basta, I.B. Barla, M. Hoffmann, G. Carle, and D. Schupke. Failure coverage in optimal virtual networks. In *Optical Fiber Communication Conf. and the Nat. Fiber Optic Engineers Conf. (OFC/NFOEC), 2013, paper OTh3E.2*, 2013.

[188] D. Schupke and R. Prinz. Performance of path protection and rerouting for wdm networks subject to dual failures. In editor, editor, *The Optical Fiber Communication Conference and Exposition (OFC) 2003, pp. 209-210*, 2003.

[189] W. Lau and S. Jha. Failure-oriented path restoration algorithm for survivable networks. In *Network operation and management symposium IEEE/IFIP, vol 1, pp. 205-218*, 2004.

[190] Wensheng He and AK. Somani. Path-based protection for surviving double-link failures in mesh-restorable optical networks. In *Global Telecommunications Conference, 2003. GLOBECOM '03. IEEE*, volume 5, pages 2558–2563 vol.5, Dec 2003.

[191] Hongsik Choi, Suresh Subramaniam, and Hyeong-Ah Choi. Loopback recovery from double-link failures in optical mesh networks. *IEEE/ACM Trans. Netw.*, 12(6):1119–1130, December 2004.

[192] Lei Guo, Lemin Li, Jin Cao, Hongfang Yu, and Xuetao Wei. On finding feasible solutions with shared backup resources for surviving double-link failures in path-protected wdm mesh networks. *J. Lightwave Technol.*, 25(1):287–296, Jan 2007.

[193] D.A Schupke, W.D. Grover, and M. Clouqueur. Strategies for enhanced dual failure restorability with static or reconfigurable p-cycle networks. In *Communications, 2004 IEEE International Conference on*, volume 3, pages 1628–1633, June 2004.

[194] H. Choi, S.S. Subramaniam, and Hyeong-Ah Choi. On double-link failure recovery in wdm optical networks. In *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 808–816 vol.2, 2002.

[195] Pankaj K. Agarwal, Alon Efrat, Shashidhara K. Ganjugunte, David Hay, Swaminathan Sankararaman, and Gil Zussman. The resilience of wdm networks to probabilistic geographical failures. *IEEE/ACM Trans. Netw.*, 21(5):1525–1538, October 2013.

[196] Hugo Patterson, Stephen Manley, Mike Federwisch, Dave Hitz, Steve Kleiman, and Shane Owara. Snapmirror: File system based asynchronous mirroring for disaster recovery. In *Proceedings of the 1st USENIX Conference on File and Storage Technologies*, FAST'02, pages 9–9, Berkeley, CA, USA, 2002. USENIX Association.

[197] W. R. Witte, M. Muhlestein, and G. Banga. Remote disaster recovery and data migration using virtual appliance migration, 2006.

[198] M.F. Habib, M. Tornatore, M. De Leenheer, F. Dikbiyik, and B. Mukherjee. Design of disaster-resilient optical datacenter networks. *Lightwave Technology, Journal of*, 30(16):2563–2573, Aug 2012.

[199] Yang Chen, Jianxin Li, Tianyu Wo, Chunming Hu, and Wantao Liu. Resilient virtual network service provision in network virtualization environments. In *Parallel and Distributed Systems (ICPADS), 2010 IEEE 16th International Conference on*, pages 51–58, Dec 2010.

[200] C. Shan-zhi, L. Xin, and W. Yan. Rmap: An algorithm of virtual network resilience mapping. In *Networking and Mobile Computing (WiCOM)*, 2011.

[201] A Jarray, Yihong Song, and A Karmouch. Resilient virtual network embedding. In *Communications (ICC), 2013 IEEE International Conference on*, pages 3461–3465, June 2013.

[202] A Jarray, Yihong Song, and A Karmouch. p-cycle-based node failure protection for survivable virtual network embedding. In *IFIP Networking Conference, 2013*, pages 1–9, May 2013.

[203] F. Gu, H. Alazemi, A Rayes, and N. Ghani. Survivable cloud networking services. In *Computing, Networking and Communications (ICNC), 2013 International Conference on*, pages 1016–1020, Jan 2013.

[204] Feng Gu, K. Shaban, N. Ghani, M. Hayat, and C. Assi. Regional failure survivability for cloud networking services using post fault restoration. In *System of Systems Engineering (SoSE), 2013 8th International Conference on*, pages 229–234, June 2013.

[205] A. Basta. Service Differentiation and Failure Coverage in Network Virtualization, Master's Thesis, Technische Universität München, 2012.