On the Trade-off between Element Availability and Cost in Virtualized Network Infrastructures

Von der Fakultät für Informatik, Elektrotechnik und Informationstechnik
der Universität Stuttgart zur Erlangung der Würde
eines Doktor-Ingenieurs (Dr.-Ing.) genehmigte Abhandlung

vorgelegt von

Sandra Herker

geb. in Ingolstadt

Hauptberichter:	Prof. Dr.-Ing. Andreas Kirstädter
Mitberichter:	Prof. Dr.-Ing. Thomas Bauschert
Tag der mündlichen Prüfung:	15. Mai 2017

Institut für Kommunikationsnetze und Rechnersysteme
der Universität Stuttgart

2017

Abstract

Lately network virtualization (technology) is receiving more and more attention. Network virtualization is the sharing of physical resources by subdividing a physical node or link into many virtual nodes or virtual links. This sharing allows to create Virtual Networks (VNs) as service-specific networks to be embedded onto a physical network in a dynamic way. Further, it is possible to share the same physical network amongst multiple network operators thereby saving on capital and operational expenditures. The calculation of the effective allocation of the physical resources (e.g. bandwidth for the connections or capacity for the servers) among the requested VNs is known as the Virtual Network Embedding (VNE) problem. One concept that uses network virtualization and recently gained a lot of attention from telecom operators and vendors is Network Functions Virtualization (NFV). It promises to virtualize entire classes of network node functions within a data-center and to deliver network services in the form of Virtualized Network Function (VNF) service chains using commercial off-the-shelf hardware and IT virtualization technologies.

Since multiple VNs or VNFs can share the physical resources of the underlying physical network (elements) in network virtualization and NFV, even a single failure in the underlying network can affect a large number of the services of the operators. The chances of a catastrophic failure could especially be higher with commercial off-the-shelf data-center hardware than in a traditional non-virtualized network infrastructure. Thus, network reliability, availability and survivability are important features for hosting high-demand services in the virtual network environment. To overcome the physical failure impact on VNs, backup and protection mechanisms are needed. However, different backup or protection mechanisms result in a different cost for the network operators. Therefore, a balance between high availability and cost has to be found. Specifically, mobile network operators require high availability (in the range of 'five nines') for offering their services to the customers, however, this also results in high infrastructure and operational costs. Therefore the important question occurs of whether it is possible to use low-cost devices in combination with network virtualization technologies to save cost and still achieve the requested availability for the services to the customers.

In light of the above, this dissertation proposes new algorithms for reliable VNE with explicit availability requirements for the services in optical transport networks and virtualized data-center networks. In the first step, a virtual network embedding algorithm which provides path protection with explicit virtual link availability constraints is developed for the use in the optical fiber wide area network. The algorithm decides when it is necessary to deploy one or more backup paths if one single path is enough to fulfill the requested availability constraint. Further, adding an optimization step results in a more resource efficient embedding. For data-

center networks, different protection backup strategies for the VNF service chains are proposed to achieve the requested service availability. An algorithm for resilient embedding of the VNF service chains in the data-center is provided, which calculates the required backups for the requested availability. Further, the algorithm for VNF service chain embedding is used to evaluate different data-center topologies consisting of mostly low-cost devices. For each of the networks, the optical fiber wide area network and the data-center network, a trade-off study is done between high availability of the service and capital expenditures. Therefore, two generic approaches are compared: (1) select only high-quality physical network elements that offer high availability and consequently demand high expenses per element, or (2) add protection capacity on the level of the VN based on lower cost components and network elements.

Based on the gained insights, conclusions can be drawn on the cost versus availability problem. From the evaluation results, it can be observed that using low-cost devices and network components as well as virtualization technologies in the different networks, it is possible to achieve a high service availability while saving cost in comparison to highly costly components. Depending on the requested service availabilities, the low-cost physical infrastructure approach can thus be cheaper. Therefore, it seems advisable to realize availability in the virtual domain with special protection mechanisms rather than in the physical domain based on dedicated hardware in real networks. However, for services requesting very high availability (more than 'five nines') often the approach of using dedicated specialized devices in combination with low-cost devices in the networks still achieves the lowest cost.

Kurzfassung

Netzvirtualisierung und Virtualisierungstechniken werden in der Internettechnologie immer mehr angewandt. Netzvirtualisierung ist die gemeinsame Nutzung der physischen Ressourcen eines Netzes durch Aufteilung eines physischen Knotens oder Links in viele virtuelle Knoten oder virtuelle Links. Diese Aufteilung ermöglicht es, virtuelle Netze (VN) als servicespezifische Netze auf einem physischen Netz dynamisch zu erzeugen. Mehrere Netzbetreiber können sich daher dasselbe physische Netz teilen und dadurch Investitions- und Betriebskosten sparen. Die Berechnung der effektiven Ressourcenzuteilung (z.B. Bandbreite für die Links oder die Kapazität für die Server) des physischen Netzes unter den verschiedenen virtuellen Netzen ist als Problem der Einbettung von virtuellen Netzen (Virtual Network Embedding, VNE) bekannt. Ein Konzept, das die Netzvirtualisierung nutzt und in der Telekommunikationsbranche seit kurzem Beachtung findet, ist die Virtualisierung spezifischer Netzfunktionen (Network Functions Virtualization, NFV). Unter NFV versteht man die Idee, die anwendungsspezifischen Funktionen von dedizierter, kostspieliger Hardware des Kommunikationsnetzes mittels Virtualisierungstechnologien auf handelsübliche Server und Switche zu transferieren. Dadurch sollen ganze Klassen von Netzfunktionen innerhalb eines Datenzentrums virtualisiert und diese dann in Form von virtualisierten Netzfunktions-Serviceketten angeboten werden.

Mittels Netzvirtualisierung und NFV können mehrere VN oder virtualisierte Netzfunktionen die Ressourcen des zugrunde liegenden physischen Netzes nutzen. Hierbei kann sich ein einziger Ausfall einer Komponente im physischen Netz auf eine Vielzahl von Diensten der Netzbetreiber auswirken. Insbesondere durch den Einsatz von Datenzentrum-Standard-Hardware könnte die Möglichkeit eines Totalausfalls höher sein als in herkömmlichen, nicht virtualisierten Netzinfrastrukturen. Deshalb sind Zuverlässigkeit, Verfügbarkeit und Ausfallsicherheit wichtige Merkmale für das Hosting von sehr anspruchsvollen Diensten in virtuellen Netzen. Um Ausfälle der VN zu vermeiden, sind Backup- und Schutzmechanismen nötig. Verschiedene Backup- und Schutzmechanismen führen jedoch zu unterschiedlichen Kosten für die Netzbetreiber. Daher muss eine Balance zwischen hoher Zuverlässigkeit, Verfügbarkeit und den Investitions-Kosten gefunden werden. Mobilfunkbetreiber fordern vor allem eine hohe Verfügbarkeit (im Bereich von 99,999% (‘five nines’)) für ihre Dienste, daraus resultieren jedoch hohe Infrastruktur- und Betriebskosten. Es stellt sich deshalb die Frage, ob es möglich ist, kostengünstige Geräte und Netzvirtualisierungstechnologien einzusetzen, um Kosten zu sparen und trotzdem die gewünschte Verfügbarkeit der Dienste zu erreichen.

Basierend auf den obigen Ausführungen werden in dieser Dissertation neue Algorithmen zur Einbettung von virtuellen Netzen mit definierten Verfügbarkeitsanforderungen für die Dienste in optischen Transportnetzen und virtualisierte Datenzentren-Netzen entwickelt. Im ersten

Schritt wurde ein neuer VNE Algorithmus für die Pfadsicherung mit explizierter Verfügbarkeitsbedingung für die Verwendung in optischen Glasfaser-Weitverkehrsnetzen entwickelt. Der Algorithmus erkennt die Notwendigkeit, ob ein oder mehrere Backup-Pfade bereitzustellen sind, um die geforderte Verfügbarkeit der Verbindungen im virtuellem Netz sicherzustellen. Ein weiterer Optimierungsschritt führt zu einer ressourcen-effizienteren Einbettung der virtuellen Netze. Für die Datenzentren-Netze werden verschiedene Backupstrategien für die virtualisierten Netzfunktions-Serviceketten entwickelt, um die geforderte Serviceverfügbarkeit zu erreichen. Ein Algorithmus, der die elastische Einbettung der virtualisierten Netzfunktions-Serviceketten im Datenzentrum und die benötigten Backups für die geforderte Verfügbarkeit berechnet, wurde entwickelt. Dieser Algorithmus wird auch verwendet, um verschiedene Datenzentrumstopologien, bestehend aus kostengünstigem Equipment, auf die Verwendbarkeit für NFV zu untersuchen. Für jedes dieser Netze, das Glasfaserkabel-Weitverkehrsnetz und das Datenzentrums-Netz, wird eine Trade-off-Studie zwischen hoher Verfügbarkeit der Dienste und den Investitionskosten durchgeführt. Zwei generische Ansätze werden deshalb verglichen: (1) die Verwendung von nur qualitativ hochwertigen physischen Netzelementen, die eine hohe Verfügbarkeit bieten und somit hohe Kosten pro Element verursachen, oder (2) die Verwendung von Backup- und Schutzkapazitäten bei virtuellen Netzen auf Basis von kostengünstigeren Komponenten.

Auf Grundlage der gewonnenen Erkenntnisse lassen sich Rückschlüsse auf das Kosten-Verfügbarkeits-Problem ziehen. Basierend auf diesen Ergebnissen wurde festgestellt, dass die Verwendung von kostengünstigen Geräten und Netzkomponenten in Verbindung mit Virtualisierungstechnologien in verschiedenen Netzen möglich ist, um eine hohe Serviceverfügbarkeit zu erreichen. Hierbei lassen sich die Kosten im Vergleich zu teuren Hardwarekomponenten reduzieren. In Abhängigkeit zur geforderten Serviceverfügbarkeit kann die Entscheidung zum Einsatz von kostengünstigerer Infrastruktur kostensenkend sein. Daher erscheint es ratsam, die Verfügbarkeit in der virtuellen Domäne mit speziellen Schutzmechanismen zu erreichen, und nicht in der physischen Domäne mittels dedizierter Hardware. Für Netze mit extrem hohen Verfügbarkeitsanforderungen (höher als ‘five nines’) ist jedoch die Verwendung spezieller Geräte in Kombination mit kostengünstigeren Geräten weiterhin notwendig, um die Kosten niedrig zu halten.

Contents

List of Figures

List of Tables

Abbreviations and Symbols

Abbreviations

3G	3rd Generation
CAPEX	Capital Expenditures
CC	Cable-Cut
COTS	Commercial Off-The-Shelf
CSPF	Constrained Shortest Path First
DC	Data-Center
DEMUX	Demultiplexer
E2E	End-to-End
ETSI	European Telecommunications Standards Institute
FIT	Failure In Time
IaaS	Infrastructure as a Service
ILP	Integer Linear Programming
IMS	IP Multimedia Subsystem
IP	Internet Protocol
ISP	Internet Service Provider
ITU-T	International Telecommunication Union - Telecommunication Standardization Sector
LB	Load Balancer
LP	Linear Programming
LTE	Long Term Evolution

MILP	Mixed Integer Linear Programming
MIP	Mixed Integer Programming
MTBF	Mean Time Between Failures
MTTF	Mean Time To Failure
MTTR	Mean Time To Repair
MUX	Multiplexer
NAT	Network Address Translation
NF	Network Function
NFV	Network Functions Virtualization
NFV-MANO	NFV Management and Orchestration
NFVI	Network Function Virtualization Infrastructure
OADM	Optical Add/Drop Multiplexer
OPEX	Operational Expenditures
OPGW	Optical Ground Wire
OTN	Optical Transport Network
OXC	Optical Cross-Connect
PIP	Physical Infrastructure Provider
PM	Physical Machine
PN	Physical Network
PSTN	Public Switched Telephone Network
QoS	Quality of Service
SLA	Service-Level-Agreement
SP	Service Provider
ToR	Top of Rack
VLAN	Virtual Local Area Network
VM	Virtual Machine
VN	Virtual Network
VNE	Virtual Network Embedding

VNF	Virtualized Network Function
VNO	Virtual Network Operator
VNP	Virtual Network Provider
VNR	Virtual Network Request
VPN	Virtual Private Network
VSCP	VNF Service Chain Placement
WDM	Wavelength Division Multiplexing

1 Introduction

This thesis addresses the topic of cost versus survivability/reliability/availability in virtualized network infrastructures. To properly introduce the work, this chapter first presents a brief overview of this space and then highlights the key motivations for the research. The need for network reliability and high availability in virtualized networks are motivated. Further reasons are identified why reliability/high service availability and cost-saving contradict and why a trade-off needs to be found. Moreover, it points out the contributions of this thesis and enumerates the author's publications towards this thesis along with an overview of the thesis chapters.

1.1 Motivation

Lately, network virtualization is gaining more and more attention. Network virtualization is the sharing of physical resources by subdividing a physical node or link into many virtual nodes or virtual links. This sharing allows to create Virtual Networks (VNs) as service-specific networks to be embedded onto a physical network in a dynamic way. Using end-to-end (E2E) virtualization, it is possible to create various service-specific networks within one operator's network. The network can be tailored to the specific needs of a service with respect to topology, routing or quality of service (QoS). Further, it is possible to share the same physical network amongst multiple network operators thereby saving on capital expenditures (CAPEX) and operational expenditures (OPEX). Multiple configurations of VNs may be created over the same physical setup. Some configurations may be more efficient than others in terms of different requirements such as optimal use of physical resources, maximizing the revenue and/or minimizing the power consumption. The calculation of the effective allocation of the physical resources (e.g. bandwidth for the connections or capacity for the servers) among the requested VNs is known as the Virtual Network Embedding (VNE) problem.

One concept that uses network virtualization and has gained a lot of attention from telecom operators and vendors recently is Network Functions Virtualization (NFV). It promises to virtualize entire classes of network node functions within a data-center and to deliver network services in the form of Virtualized Network Function (VNF) service chains using commercial off-the-shelf hardware (COTS) and IT virtualization technologies. However, availability becomes an important issue when purpose-built telecom hardware designed for the 'five nines' standard via built-in failure protection and recovery mechanisms is replaced by the COTS hardware. With COTS data-center hardware, failure probabilities could be higher than in traditional physical network infrastructure.

Since in network virtualization and NFV multiple VNs or VNFs can share the same physical resources of the underlying physical network (elements), even a single failure in the underlying network can affect a large number of the services of the operators. Therefore, survivability, network reliability and high (service) availability are important features in virtualized environments. For achieving this survivability and high availability of the services or VNs, several resilience and backup mechanisms have been investigated. Specifically, mobile network operators require high availability (in the range of 'five nines') for offering their services to the costumers, however, this also results in high infrastructure and operational costs. Therefore, one of the main requirements for the operators when building communication networks is to reduce cost while fulfilling the customers' requirements. One important question, therefore, is how much should be paid for this reliability/high availability of the services. Furthermore, how can the operators save on cost and still reach the requested high availability for their services? Is it possible to use low-cost devices in combination with network virtualization technologies to save on cost and achieve the requested service availability?

1.2 Problem Description and Main Contributions

This work is motivated by the need for reliability and availability and for cost saving for network operators in a virtualized network environment.

Network reliability, availability and survivability are important features for hosting high-demand services in virtual network environments. Especially in virtualized environments and cloud environments, the reliability and availability issue plays an important role due to the fact that the chances of a catastrophic failure could be higher than in a traditional non-virtualized network infrastructure. Since multiple VNs can share the physical resources of the underlying substrate, even a single failure in the substrate can affect a large number of VNs and the services they offer. To overcome the physical failure impact on VNs, backup and protection mechanisms are needed. However, different backup and protection mechanisms result in different cost for the network operators. Therefore, a balance between high reliability/availability and low cost has to be found.

To address these issues in this work, new solutions for reliable VNE are developed and a trade-off study is done between availability and cost in virtualized network environments. Two different network types, the optical transport network and the data-center network, are chosen for the study since optical transport networks and data-center networks are important and integral parts of an E2E telecommunication network. To minimize cost, network operators need to consider the required network availability already at the network design stage. One generic approach to reach the availability target is to select only high-quality physical network elements that offer high availability and consequently demand high expenses per element. The other approach to achieve high availability is to add protection capacity on the level of the virtual network based on low-cost components. To evaluate this problem of the trade-off between expensive equipment and protection in the virtualized environment, new special algorithms and methods are developed.

In the first part of the cost versus availability trade-off problem, optical transport networks will be examined. A new VNE algorithm with explicit availability constraint for path protection

is developed. In the second part of the cost versus availability trade-off problem, the focus goes deeper into the network structure at the end points of the optical transport network: the data-centers. By connecting the optical transport and the data-center network, a complete high-available E2E communication network can be created.

The main contributions of this thesis are summarized as follows:

- Novel virtual network embedding algorithms:
  - for path protection with an explicit availability constraint,
  - for Virtualized Network Function service chain protection with high availability constraints.
- Cost model design for different networks:
  - for optical transport networks (fiber networks),
  - for data-center networks.
- Evaluation of different data-center topologies for the suitability of Virtualized Network Function service chains.
- Experimental findings with simulations: Trade-off between cost and (element) availability in different virtualized network infrastructures
  - for optical transport networks,
  - for data-center networks.

The results of this research towards this thesis have been published in the proceedings of various conferences and workshops [1, 2, 3, 4, 5, 6]. In addition, based on this research a patent has been filed [7, 8].

1.3 Thesis Outline and Organization

This thesis is subdivided into three main parts. The first part comprises the fundamentals on the background and state-of-the-art on network virtualization, virtual network embedding, and reliability and availability in Chapter 2. The second part introduces the problem of the trade-off between cost and availability in optical transport networks in Chapter 3 and provides details on the mechanisms, algorithms and evaluation. The third part consists of Chapter 4, which evaluates the cost versus availability problem in data-center networks and the corresponding mechanisms and algorithms.

Chapter 2 first introduces the fundamentals of network virtualization. Besides a definition and use case for network virtualization, the concept of Network Functions Virtualization (NFV) is explained. The fundamental concepts of optical transport networks and data-center networks

are presented. This chapter also introduces fundamental reliability and availability terms. Further, failure analysis is addressed for the different networks, optical transport network and data-center network. Finally, the state-of-the-art in virtual network embedding (VNE) is addressed by first introducing the concept, and second by providing an overview on well-known algorithms. Several VNE algorithms in different areas are presented and compared with a highlight on algorithms for reliability. Parts of this chapter are contained in all publications mentioned earlier, although a few papers provide more details on network virtualization use cases [1] and VNE and VNE algorithms [2].

The purpose of Chapter 3 is to discuss the trade-off between cost and availability in optical fiber transport networks. It starts with the modeling of the optical fiber transport network and its cost and the requested virtual networks. Further, a cost model for deploying fiber in relation to MTBF has been designed. The major part of the chapter presents a new VNE algorithm for path protection with explicit availability constraints. First the problem of VNE with explicit availability constraints is formulated mathematically. In the next step, a heuristic is developed to solve the mathematical problem in polynomial time. Next, the VNE algorithm is combined with the cost model to answer the question about the trade-off between cost and (element) availability in optical fiber transport networks. The chapter concludes with a discussion of the results and recommendations for network operators. The content of this chapter has already been published in two papers [3, 4].

Chapter 4 discusses the problem of finding a suitable data-center architecture for VNF services while saving cost and achieving a requested reliability of services. First the (virtualized) data-center network and VNF service chain are described and modeled. Next, for comparing the cost between these data-centers, a cost model is designed which considers the switch and server costs. New algorithms for placing the VNF service chains in the data-centers with a defined availability are developed and examined. These algorithms are used to examine the suitability of the data-center topologies. Finally, conclusions are drawn for achieving a low-cost and suitable data-center for the VNF service chains requesting high service availability. The topics and content of this chapter have been published in two papers [5, 6].

Chapter 5 concludes the work presented in this thesis and gives an outlook on further work items.

2 Background

This chapter introduces fundamental concepts that underpin the work presented in this thesis. These are network virtualization and virtual network embedding, reliability and availability.

Section 2.1 introduces network virtualization and its concept and use cases. Definitions of reliability and availability are given in Section 2.2 followed by covering the fundamentals of the used specific networks in this thesis, the optical fiber transport network in Section 2.3 and the data-center network in Section 2.4. A failure analysis in fiber and data-center networks is presented in Section 2.5. Section 2.6 introduces the basic concept of virtual network embedding (VNE) and its terminology used within this thesis. First, the general VNE problem and the reliable/survivable VNE problem is explained in detail. Further, selected algorithms of VNE are explained with a focus also on reliability and availability.

Several parts of this chapter – especially of Section 2.1.3 and Section 2.6 – have been published in [1] and [2] respectively.

2.1 Network Virtualization

This section presents an overview of the network virtualization paradigm. It starts with a definition followed by design goals of network virtualization. The different stakeholders and their roles in network virtualization are introduced. Further, various use cases for network virtualization especially in the field of mobile communication are provided. The last part explains the concept of network functions virtualization.

2.1.1 Overview and Definition

2.1.1.1 Definition

Network virtualization can be defined as a technique for isolating computational and network resources for creating multiple independent and programmable virtual networks (VNs). Therefore, multiple networks and services can be implemented in isolated logical networks on top of a single shared physical infrastructure [9]. Two basic components of network virtualization exist: link virtualization and node virtualization [10]. Link virtualization enables the transport of multiple separate virtual links over a shared physical link. A virtual link is often identified explicitly by a tag. However, it can also be identified implicitly, by a time slot or a wavelength.

Node virtualization is based on isolation and partitioning of hardware resources of a physical node (e.g. CPU, memory, storage capacity) into slices. Each slice is allocated to a virtual node according to the requirements. VNs can be created combining the virtualization of physical nodes and of the links interconnecting those physical nodes.

2.1.1.2 Design Goals

There are several design goals for realizing network virtualization. The main goals are programmability, (resource) isolation, flexibility, quick reconfiguration, scalability, manageability, network abstraction, topology awareness, stability and convergence, heterogeneity and legacy support [9, 11, 12]. Isolation can be achieved with server virtualization like Xen [13], and with a flowvisor and a flow table separation [14]. Network isolation refers to guaranteeing complete separation between the VN components over the same physical network. This means that overuse of resources in one network must not affect the service quality in the other network.
However, there are also several research challenges for realizing network virtualization: resource and topology discovery, resource allocation or VNE, creation of virtual nodes and virtual links, admission control and usage policing, interfacing, signaling and bootstrapping, naming and addressing, mobility management, monitoring/configuration and failure handling, security and privacy and interoperability issues. In this thesis, one of the important challenges to face is the problem of resource allocation, especially the VNE problem.

2.1.1.3 Historical Perspective

Network virtualization can be seen historically coming from different concepts: Virtual Local Area Networks (VLANs), Virtual Private Networks (VPNs), active and programmable networks, and overlay networks [11, 9].

A Virtual Local Area Network (VLAN) is a logical group of networked hosts that appear to be on the same LAN despite their geographical distribution. Network administration, management and reconfiguration of VLANs are easier than in their physical counterparts as VLANs are based on logical connections. Further, VLANs provide high levels of isolation [11].

A Virtual Private Network (VPN) is a private network that is built over shared or public communication networks like the Internet. Security mechanisms (e.g. encryption) allow VPN users to securely access the network from different locations [11].

Active and programmable networks should create, deploy, and manage novel services on the fly in response to user demands. Programmability of network elements is the capability to change their behavior through possibly remote instructions [11].

Overlay networks are the most recent ancestor of network virtualization. An overlay network is a logical network created on top of one or more physical networks. Overlays do not demand any changes to the underlying network. A fundamental difference between overlay network and network virtualization is that overlay networks are realized through virtualization of computational resources at the network edges and also are unable to achieve complete isolation of

network resources. However, network virtualization intends to isolate computational resources inside the network as well as network resources between them [9, 11].

2.1.2 Business Roles in Network Virtualization

Compared to the traditional networking model, the roles in the network virtualization model are different. In the traditional Internet model, the major actors are service providers and Internet Service Providers (ISPs). An ISP offers customers the access to the Internet through its own infrastructure or infrastructure from other ISPs while service providers offer services on the Internet. In the network virtualization environment the following roles exist [10, 12, 15]:

- The **Physical Infrastructure Provider (PIP)** deploys and runs the physical network resources. Using virtualization technology, the physical network is partitioned into isolated virtual slices by the PIP. These resources are offered to another service provider, and not to end users. However, a PIP customer might as well be a corporation using the VN for its internal use rather than to build end user services. The PIP has a view of the resources allocated to each VN, however; the protocols running inside are unknown to the PIP.
- The **Virtual Network Provider (VNP)** is responsible for finding and composing the adequate set of virtual resources from one or more PIPs, in order to fulfill the virtual network operator request. The VNP leases slices of the virtualized infrastructure from one or more PIPs and puts them together. The VNP does not provide a ready network to the virtual network operator yet, just an empty container where the virtual network operator builds the protocols that compose the VN.
- The **Virtual Network Operator (VNO)** deploys any protocol stack and network architecture over a VN, independent of the underlying physical network technologies. The VNO operates, maintains, controls and manages the VN. Ideally, the fact that resources are virtual rather than physical should not imply any major impact from an operational point of view. Thus, the role of the VNO should be indistinguishable from that of any operator running a native network infrastructure. VNOs have a unified view of the network, regardless of the multiple infrastructure domains on which it is built.
- The **Service Provider (SP)** uses the VNs to offer its services to end users.

2.1.3 Use Cases for Network Virtualization

This section presents various use cases for network virtualization. From each use case a set of important requirements is derived. More examples of use cases can be found in [1].

2.1.3.1 Network Sharing

Operating and managing an E2E mobile network involves significant costs. OPEX and CAPEX can be reduced on the infrastructure using network sharing [16]. Using virtualization, isolated

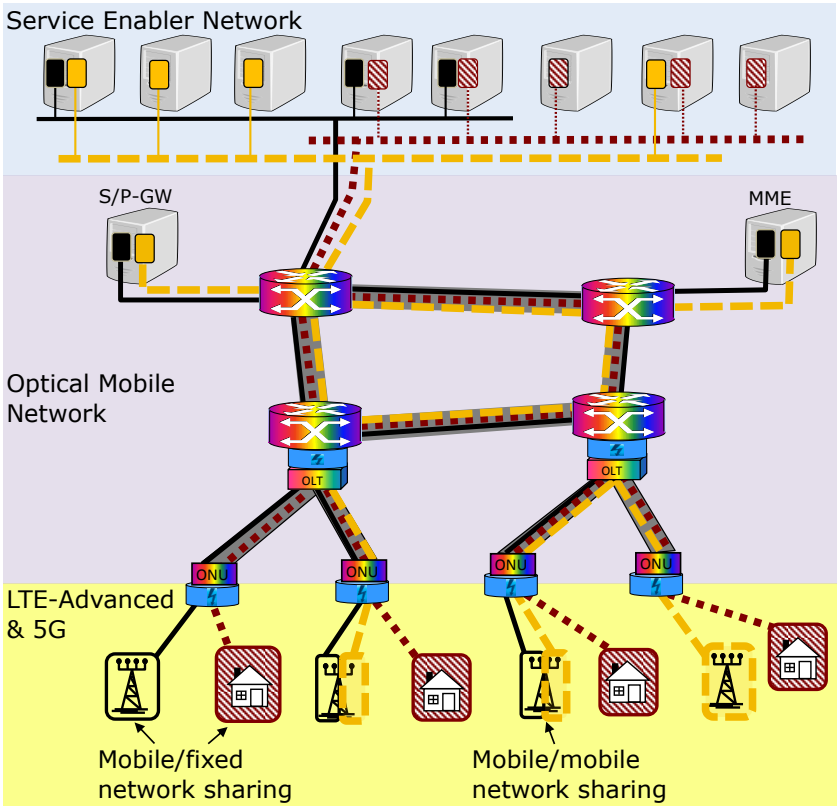

Figure 2.1: Network sharing use case scenario [1]: The three different isolated networks are E2E virtual networks, which share the underlying physical infrastructure.

VNs can be created that multiple operators can use. The infrastructure may not necessarily belong to the operators but to a federation of smaller physical infrastructure owners. In this way global mobile networks can be realized.

Figure 2.1 presents three different isolated networks. The black line and dashed line show E2E virtual mobile networks. The network with dotted lines shows an E2E virtual fixed network. From the operational point of view, the VNs must be independent of each other. Sharing resources in this way can significantly reduce OPEX and also facilitate a greener operation. An example that demonstrates this point is the fixed network connectivity using fiber networks and fiber access to the households. Fiber access implies the ability to support bandwidth far beyond the needs of a single household. In fact, one wavelength can support over 10 Gbps of data, enough to support a base station. This implies that mobile network operators can re-utilize the fixed network as a back-haul, saving considerable CAPEX. Requirements are the complete E2E isolation of the different networks, security and privacy and a dynamic and flexible sharing. Further standardized interfaces have to be provided for communication between provider and PIPs and among multiple different PIPs.

A special case would be the service-specific network, which is the creation of various service-specific VNs within the network of a single operator. The network can be tailored to the specific needs of a service with respect to topology, addressing, routing, QoS, caching, processing. An additional requirement here is the mapping of specific requirements of the service.

2.1.3.2 *Disaster Recovery*

A VN which is easily reconfigurable can be useful in saving lives during disasters like earthquakes. Figure 2.2 shows a scenario for disaster recovery using network virtualization in geographically distributed clouds. With tsunami-like and earthquake-like disasters, there may be sufficient warning time in which the network can be dynamically reconfigured. Essential services can be migrated to safer areas and the links dynamically reconfigured to maintain network connectivity, providing maximum possible coverage and uptime. Networks amongst various operators can also be consolidated temporarily, replacing the destroyed resources of one operator by sharing with the existing resources of another competing operator. This in turn may help save countless lives. Requirements are auto-(re)configuration and virtual server migration between geographically distributed clouds.

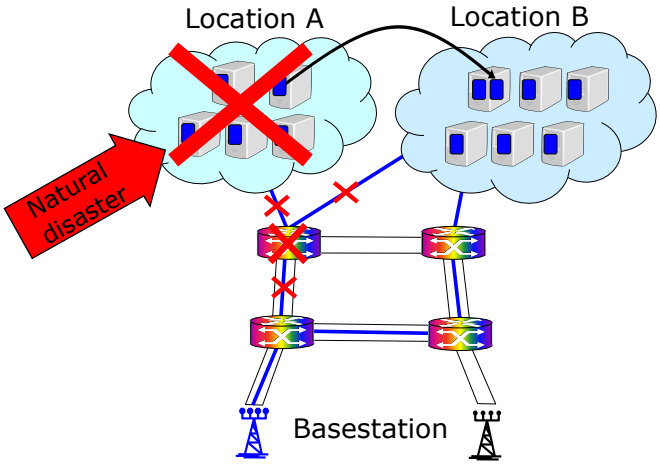

Figure 2.2: Disaster recovery use case scenario [1]: Essential services can be migrated to a geographically distributed cloud.

2.1.3.3 *Multi-Generation Network*

In a multi-generation network, different cellular versions like 3G and Long Term Evolution (LTE) can run in parallel on the same physical network. Therefore, separate physical networks will not be necessary. The operators can save infrastructure and operational costs by having only one physical network hosting several VNs on it. Two virtual base stations can be created on a single physical base station, one supporting LTE and the other 3G. Here the important requirements are isolation between the different networks and reliability.

2.1.3.4 *Green Network*

The ability to seamlessly migrate network functionalities from one physical host to another has another advantage. During low demand periods a number of machines can be shut down and the functionalities consolidated to a smaller set of physical hosts. Shutting down machines helps in energy conservation resulting in a greener network operation. Requirements for this use case are energy-efficient resource allocation and VM migration techniques.

In order to realize the requirements of the above use cases in real systems a number of technical challenges need to be overcome. Complete E2E isolation is difficult to achieve. The different isolation techniques for virtualization of the various parts of the network may occur at different network layers. Security and privacy for one operator cannot be fully guaranteed between coexisting VNs. Only a certain level of security among the isolated networks through secured tunnels or encryption is possible as the physical layer/resource could be compromised directly and affect multiple VNs at the same time. For efficient resource allocation a suitable VNE algorithm needs to be investigated. The embedding should be done in a way that the utilization of the physical resources is maximized.

2.1.4 Network Functions Virtualization

One important application of virtualization is **Network Functions Virtualization (NFV)**.

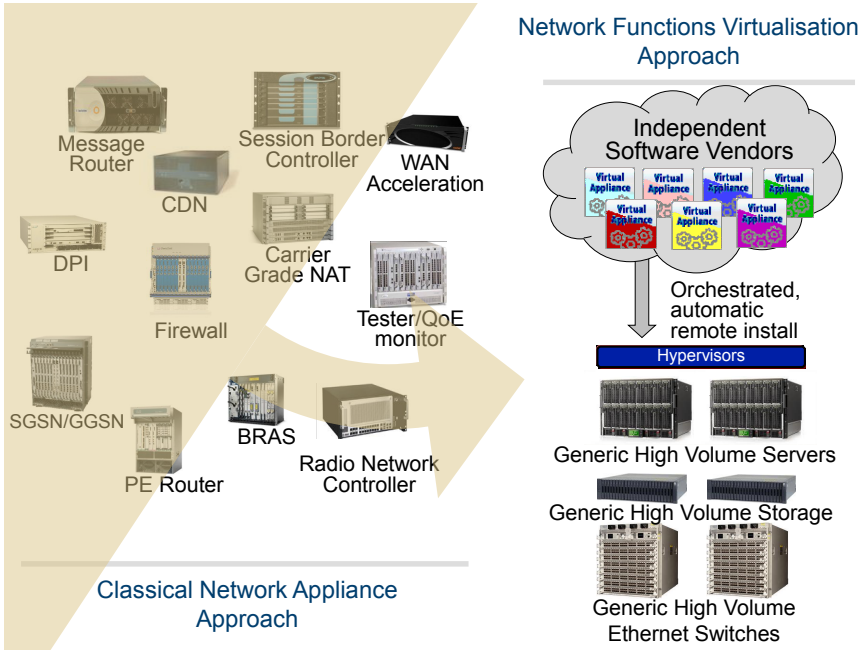

Figure 2.3: Network Functions Virtualization [17]

The European Telecommunications Standards Institute (ETSI) defines it as follows:

> "Network Functions Virtualization (NFV) aims 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 data-centers, network nodes and in the end user premises. It involves the implementation of network functions in software that can run on a range of industry standard server hardware, and that can be moved to, or instantiated in various locations in the network as required without the need for installation of new equipment." [17]

The NFV approach is shown in Figure 2.3.

The concept of NFV is an emerging network architecture concept to virtualize network functions (NFs) like firewalls, gateways or carrier-grade Network Address Translation (NAT). These functions are implemented as software on industry standard high volume servers, switches and storage using IT virtualization technologies into building blocks. These blocks may be connected together in the form of Virtualized Network Function (VNF) service chains to create network services.

2.1.4.1 Benefits, Challenges and Design Criteria for NFV

The benefits of NFV are reduced equipment cost and power consumption through consolidating equipment and exploiting the economies of scale of the IT industry. Furthermore, the time to deploy new networking services can be reduced to support changing business requirements.

The NFV approach can be applied by telecom operators to operate telco-related services. Such an approach promises to reduce CAPEX and OPEX for service deployment. Moreover, it can also reduce the time to market. It delivers agility and flexibility by quickly scaling up or down services to address changing demands. Further, innovation is supported by enabling services to be delivered as software on any industry-standard server hardware. The availability of network appliance multi-version and multi-tenancy allows the use of a single platform for different applications, users and tenants. This allows network operators to share resources across services and across different customer bases. Targeted service introduction based on geography or customer sets is possible. Services can be rapidly scaled up/down as required [18, 19].

To leverage these benefits there are a number of technical challenges which need to be addressed. Examples include achieving high performance virtualized network appliances which are portable between different hardware vendors, and with different hypervisors, and the coexistence with existing hardware-based network platforms. The integration of the third-party, open-source software and the commercial off-the-shelf (COTS) hardware and software components into a system is challenging. Further, the management and orchestration of many virtual network appliances is difficult and NFV will only scale if all of the functions can be automated. It must ensure the appropriate level of resilience to hardware and software failures and integrate multiple virtual appliances from different vendors [18, 19].

The challenges to face for reliability in NFV and especially in VNFs is the increasing system complexity with the addition of virtualization. In addition, user inexperience in operating the virtual environment can cause problems. Since VNF failure will impact E2E service reliability and availability, a VNF should provide reliability as high as the one of the non-virtualized NF. Therefore, VNF failures should never impact other applications, hardware failures should only affect those VMs assigned to that specific hardware and connectivity failures should only affect connected NFs, etc. All resiliency mechanisms should be designed for a multi-vendor environment. Different reasons exist for VNF failures compared to non-virtualized NF like lower availability of VNF than non-virtualized, different hardware (e.g. commodity-grade hardware) or the existence of multiple software layers (hypervisor, guest operating system) and the fact that failures of the underlying hardware and software infrastructure can also affect the VNF. Compared to non-virtualized NFs new failure modes arise like failures on the hypervisor level that did not previously exist in the box-model (i.e. proprietary hardware network boxes) since

several VNFs can be hosted on the same physical host. Further simultaneous failures of multiple VNF components can occur due to a failure of the underlying hardware [20, 19, 21].

Therefore, the important design criteria for virtualization of NFs for reliability are:

- Service continuity: E2E availability of the telecommunication services.
- Failure containment: automated recovery from failures, prevention of single point of failure in the underlying architecture.
- Suitability in a multi-vendor environment or hybrid infrastructure.

2.1.4.2 NFV Framework

The NFV framework consists of three main components [22]:

Virtualized Network Functions (VNFs) are software implementations of NFs that can be deployed on a network functions virtualization infrastructure. A VNF may consist of one or more VMs. Instead of having customized hardware appliances for each NF, a VNF can run different software and processes on top of standard high-volume servers and switches or even cloud computing infrastructure.

Network Functions Virtualization Infrastructure (NFVI) is the entirety of all hardware and software components that build the environment in which VNFs are deployed. The NFVI can span several locations. The network providing connectivity between these locations is regarded as part of the NFVI.

Network Functions Virtualization Management and Orchestration (NFV-MANO) architectural framework is a functional block in the NFV architecture framework, which stands for management and orchestration. It is required for provisioning of the VNF and for performing operations like configuration, management of the VNFs and the infrastructure these functions run on. Further, it manages the NFVI and orchestrates the allocation of the resources needed by the VNFs. Orchestration describes the automated arrangement, coordination and management of NFVI and VNFs.

2.1.4.3 Use Cases

Use Cases for NFV can span a wide range from NFV Infrastructure as a Service (NFVIaaS), VNF as a Service (VNFaaS) or Virtual Network Platform as a Service (VNPaaS), virtualization of mobile base station, the home environment or mobile core network and IMS and fixed access network functions virtualization [23].

One interesting use case for this thesis is the *Virtualization of Mobile Core Network and IMS*. The virtualization of a mobile core network is targeting at a more cost efficient production environment. It allows network operators to cope with the increasing traffic demand in mobile networks. Further, benefits lead to better resource utilization (including energy savings), more

flexible network management (no need to change hardware for nodes' upgrades), hardware consolidation, easier multi-tenancy support and faster configuration of new services [23]. This use case states clear requirements for high service availability.

2.2 Reliability and Availability

In this section the fundamentals of reliability and availability are explained. Important terms for reliability and availability are defined, like MTBF, MTTR and so on.

2.2.1 Definition: Reliability and Availability

Reliability: The reliability is the probability that the system operates successfully for a given period of time under environmental conditions [24].

Resilience: Resilience is the ability of the system/network to provide and maintain an acceptable level of service in the face of various faults and challenges to normal operation [25].

Survivability: Survivability is the capability of a system/network to provide its service, in a timely manner, in the presence of threats such as attacks or large-scale natural disasters [25].

Availability: Availability is the probability that an item is up at any point in time and will be able to perform its designed functions [24].

The availability of the system is a measure of how much of the operating time the system is up.

The availability A can be quantified via the Equation (2.1) as system uptime divided by the sum of system uptime and system downtime [24].

$$A = \frac{Uptime}{Uptime + Downtime} \tag{2.1}$$

Further reliability parameters are Mean Time Between Failures (MTBF), Mean Time To Repair (MTTR) and Mean Time To Failure (MTTF).

2.2.2 Definition: MTBF, MTTR, MTTF and FIT

Mean Time Between Failures (MTBF) is the average time between two occurrences of a failure in a network element or component [26]. Further, it is a reliability term used to provide the amount of failures per million hours for a product or device [27].

Mean Time To Repair (MTTR) is the time needed to detect the failure, repair the failed network element or component and return it to normal operations [26]. In general, in an operational system, repair means replacing a failed hardware component [27].

Mean Time To Failure (MTTF) is the mean time a device or equipment is expected to last in operation. It is a measure of the reliability for non-repairable items. MTTF is a statistical value and is measured over a long time period and a large number of units. Technically, MTBF should be used for repairable items, while MTTF should be used for non-repairable items [27, 28].

Failure In Time (FIT) is another way of describing MTBF. The FIT rate of a device is the number of failures that can be expected in one billion hours of operation for a device [27].

$$MTBF[h] = 10^9 h/\,FIT \tag{2.2}$$

As mentioned above the availability A of any component is defined as the percentage of time when the component is operational and fulfilling its requirements, i.e. as the relation between its uptime and the sum of its uptime and downtime [24]. Using common parameters as MTBF and the MTTR [29] the availability A can be expressed by

$$A = \frac{MTBF \text{ - } MTTR}{MTBF} \tag{2.3}$$

(Service) Availability is generally expressed in terms of 'number of nines' (9's): The more 'nines' the less downtime of the service in minutes per year and higher availability. In Table 2.1 the service availability and its downtime are given.

Table 2.1: Service availability and downtime [26]

Number of 9's	Service Availability (%)	System Type	Annualized Down Minutes	Practical Meaning
1	90	Unmanaged	52,596.00	Down 5 weeks per year
2	99	Managed	5,259.60	Down 4 days per year
3	99.9	Well managed	525.96	Down 9 hours per year
4	99.99	Fault tolerant	52.60	Down 1 hour per year
5	99.999	High availability	5.26	Down 5 minutes per year
6	99.9999	Very high availability	0.53	Down 30 seconds per year
7	99.99999	Ultra availability	0.05	Down 3 seconds per year

2.3 Fundamentals of Optical Fiber Transport Networks

In this section, the fundamentals of the fiber transport network are explained. The optical fiber network plays an important role in this thesis and is used as the underlying physical network in the virtualized environment in Chapter 3.

2.3.1 Definition

An optical transport network is a network that transmits information over optical media. Optical transport networks are used to connect a large group of users spread over a geographical area. ITU-T defines an Optical Transport Network (OTN) as "a set of optical network elements connected by optical fiber links, able to provide functionality of transport, multiplexing, switching, management, supervision and survivability of optical channels carrying client signals" [30].

OTN was designed to provide support for optical networking using Wavelength-Division Multiplexing (WDM). WDM is an optical multiplexing technology for utilizing the huge bandwidth capacity in optical fibers. The basic principle is to split the bandwidth of an optical fiber into a number of non-overlapping subbands or optical channels. Multiple optical signals are then transmitted simultaneously and independently in different optical channels over a single fiber, each signal being carried by a single wavelength. The huge bandwidth can be divided up into a number of optical channels with each channel operating at any feasible rate (e.g. peak electronic speed of up to hundreds of gigabits per second). Theoretically a single fiber is capable of supporting over 1000 optical channels or wavelengths at up to hundreds of gigabits per second. As network medium, a simple fiber link, a passive star coupler, or any type of optical network can be used [31, 32].

WDM networks can be classified into two broad categories: broadcast-and-select WDM networks and wavelength-routed WDM networks.

A broadcast-and-select WDM network is a WDM network that shares a common transmission medium and employs a simple broadcasting mechanism for the transmission and receiving optical signals between network nodes [31].

A wavelength-routed WDM network is a WDM network that employs wavelength routing to transfer data traffic. It typically consists of routing nodes interconnected by WDM fiber links. Each routing node employs a set of transmitters and receivers for transmitting signals to and receiving signals from fiber links and an optical cross-connect (OXC) to route and switch different wavelengths from an input port to an output port. Each fiber link operates in WDM and supports a certain number of optical channels or wavelengths. In a wavelength-routed (wide-area) WDM network, a pair of network nodes communicates through an E2E optical connection that may consist of one or more all-optical connections called light paths. A light path is a unidirectional all-optical connection between a pair of network nodes. It can span multiple fiber links and use one or multiple wavelengths. Two light paths cannot share the same wavelength on a common fiber link (referred to as the wavelength-distinct constraint). However, on different fiber links two light paths can use the same wavelength (referred to as the wavelength-reuse property). If there is no wavelength conversion possible, a light path must use the same wavelength on all

the links it spans (referred to as the wavelength-continuity constraint). To eliminate this constraint wavelength converters at network nodes to provide wavelength conversion capability in the network can be used [31].

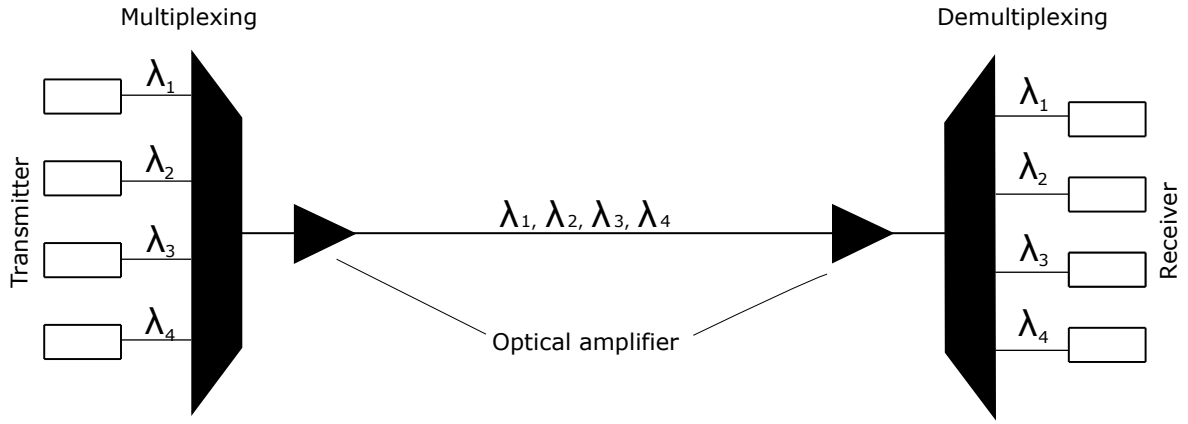

Figure 2.4: A WDM transmission system

Figure 2.4 shows a block diagram of a WDM transmission system.

2.3.2 Components

In the following the components of a WDM system are explained:

Optical fiber: Optical fiber is an excellent physical medium for high-speed transmission. It can provide extremely low-attenuation transmission over a huge frequency range and thus has a number of advantages over traditional transmission media such as copper and air. An individual single-mode fiber can provide a transmission bandwidth of about 50 Tbps. An optical fiber consists of a fine cylindrical glass core surrounded by a glass cladding. A multi-mode fiber is an optical fiber with many different guided rays propagating inside the core. For single-mode fiber, only one mode exists in which a light ray can propagate and the light propagates in a straight direction along the fiber. Therefore, data can be transmitted at up to hundreds of gigabits per second over hundreds of kilometers without any amplification in a single-mode fiber [31].
The external factors and the installation type of the cable in the telecommunication network are the basic requirements for determining the structure, the dimensions and the materials of an optical fiber cable. The main components of an optical cable can be divided into the following five groups: 1) optical fiber coatings, 2) cable core, 3) strength members, 4) water-blocking materials (if necessary), 5) sheath materials (with armor if necessary). The cable sheath protects the cable core from mechanical and environmental damage [30].

Optical fiber couplers: Coupler is a general term that covers all devices that combine light into or split light out of a fiber. Optical fiber couplers can be either active or passive devices.

Optical amplifiers: Generally the transmission distance of optical fiber systems is limited by fiber attenuation and by fiber distortion. An approach to compensate this loss is the use of

optical amplifiers. An optical amplifier is used to amplify the power of an optical signal in an optical transmission system [30, 31].

Fiber-Optic Transmitters and Receivers (Transceivers): Fiber-optic transmission systems consist of a transmitter on one end of a fiber and a receiver on the other end. Often a 'transceiver' is used which includes both transmission and receiver in a single module.
Optical transmitters: The role of the optical transmitter is to convert an electrical signal into a corresponding optical signal and launch the resulting optical signal into the optical fiber. The optical transmitter consists of the following components: optical source (laser), electrical pulse generator and optical modulator [30, 31].
Optical receivers: The receiver converts the optical signal back into electrical form and recovers the data transmitted through the optical system. The receiver consists of a detector (e.g. photodiode) that converts light into electricity through the photoelectric effect [31, 30].

Optical wavelength MUX/DEMUX: A wavelength multiplexer (MUX) is a branching device with two or more input ports and one output port where the light in each input port is restricted to a pre-selected wavelength range and the output is the combination of the light from the input ports. A wavelength demultiplexer (DEMUX) is a device which performs the inverse operation of a wavelength multiplexer, where the input is an optical signal comprising two or more wavelength ranges and the output of each port is a different pre-selected wavelength range [30]. A WDM system uses a MUX at the transmitter to join the signals together and a DEMUX at the receiver to split them apart. With the right type of fiber it is possible to have a device that does both simultaneously and can function as an optical add/drop multiplexer [31].

Switching elements: An Optical Cross-Connect (OXC) is a device used by telecommunications carriers to switch high-speed optical signals in a fiber-optic network. In a wavelength-routed WDM network, an OXC can switch the optical signal on a WDM channel from an input port to an output port without any optoelectronic conversion of the signal [32]. An Optical Add/Drop Multiplexer (OADM) consists of a DEMUX, followed by a set of 2 x 2 switches (one switch per wavelength) followed by a MUX. An OADM can be viewed as a special case of an OXC [31, 30].

Transponders: In order to allow coexistence of equipment from different vendors at the border of WDM optical transmission, optical transponders are used [30].

2.4 Fundamentals of Data-Center Networks

In this section the fundamentals of the data-center (DC) network are explained. DC networks play an important role in this thesis and are used as the underlying physical network in the virtualized environment for the VNF service chain embedding in Chapter 4.

2.4.1 Definition

A data-center (DC) is a facility optimized for hosting computer systems and associated components, such as telecommunications and storage systems. It generally includes redundant

or backup power supplies, redundant data communications connections, environmental controls and various security devices. It can have one or more connections to the public Internet, often via redundant and physically separated cables into redundant routers. It consists of servers/physical machines (PMs), storage and network devices (e.g. switches, routers, and cables), power distribution systems and cooling systems [26, 33, 34, 35, 36].
The DC network holds a central and important role in a DC as it interconnects all of the DC resources together. A DC network is described by the network topology, routing/switching equipment and the used protocols (e.g. Ethernet and IP). A DC network is traditionally set up as a multi-root spanning-tree topology comprising different types of devices such as routers, switches, load balancers, and firewalls. DC networks need to be scalable and efficient to connect several thousands or even hundreds of thousands of servers (especially for cloud computing) to handle the growing demands [26, 33, 34, 35, 36].

A *virtualized data-center* is a physical DC where a number of or all of the hardware (e.g. servers, routers, switches and links) are virtualized [36]. A *virtual data-center* is a collection of virtual resources (VMs, virtual switches and virtual routers) connected via virtual links. It is a logical instance of a virtualized DC consisting of a subset of the physical DC resources [36].

2.4.2 Different Data-Center Topologies

The following DC topologies are considered for the analysis in this thesis. These topologies can be categorized in switch-only or switch-centric topologies like 2-/3-tier tree and Fat-Tree, and hybrid or server-centric topologies like BCube and DCell. In switch-only or switch-centric architectures the interconnection and routing intelligence are done by switches and packet forwarding is implemented exclusively using switches. In hybrid or server-centric architectures packets are forwarded using a combination of switches and servers.

2.4.2.1 Two-tier Tree Topology

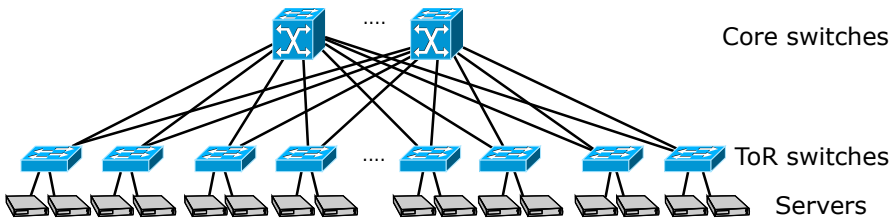

Figure 2.5: 2-tier tree architecture

Two-tier (Figure 2.5) and three-tier architectures are typical tree-based DC topologies. The two-tier architecture is a simple architecture with a top of rack (ToR) tier and a core tier. Each core switch is connected to all ToR switches, resulting in a fully meshed structure. Access switches for server connectivity are collapsed to high-density switches which provide the switching and routing functionality for access switching interconnections and the various servers.

The advantages of the two-tier architecture are the simple topology due to fewer switches and managed nodes compared to the three-tier architecture, reduced network latency by reduced

number of switch hops and lower aggregated power consumption. However, the disadvantages are limited scalability compared to the three-tier architecture as the core switches need to offer high port counts – thus often becoming a bottleneck of the DC [34, 33].

2.4.2.2 Three-tier Tree Topology

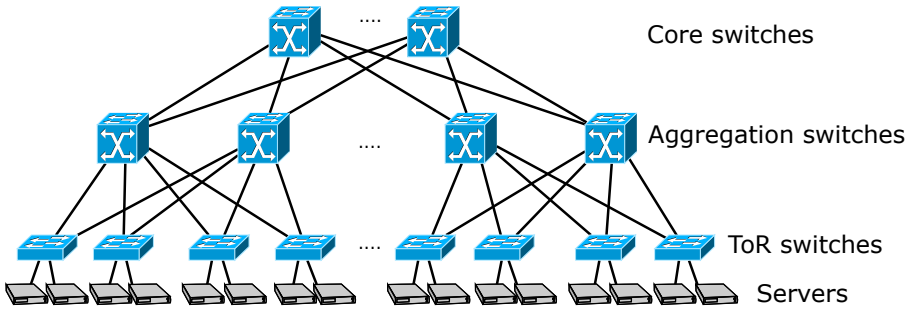

Figure 2.6: 3-tier tree architecture

The three-tier architecture (Figure 2.6) follows a multi-rooted tree network topology composed of three tiers of network switches: access, aggregate, and core. At the access tier, each server connects to a ToR switch. Each ToR switch connects to two switches (one as primary and one as backup) at the aggregation tier. Each aggregation switch connects with all switches at the core tier. The core switches provide routing to and from the enterprise core network. The three-tier design is based on a hierarchical design, so its main benefit is scalability. One could add new aggregation switch pairs with no need to modify the existing aggregation pairs. The disadvantages of three-tier design are higher latency due to the additional layer, additional congestion/oversubscription in the design (unless bandwidth between nodes is dramatically increased), more managed nodes (adding a certain amount of complexity for operation and maintenance), higher energy consumption and the need for additional rack space [34, 33].

2.4.2.3 Fat-Tree

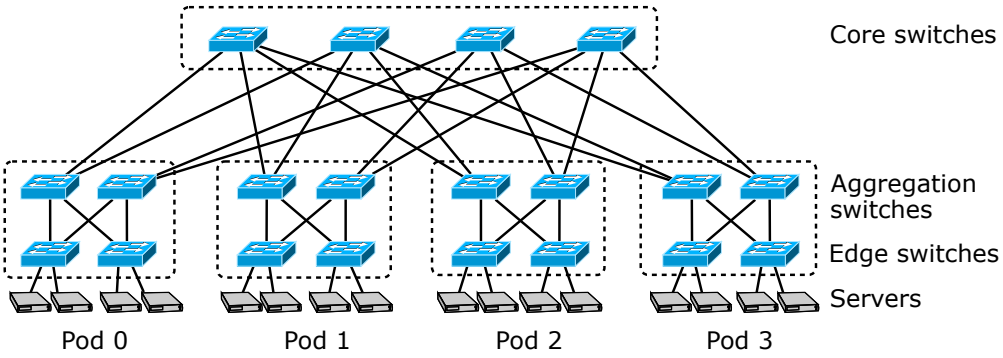

Figure 2.7: Fat-Tree architecture

The Fat-Tree topology (Figure 2.7) is a three-tier architecture that uses Clos topology [33]. Fat-Tree overcomes the problem of the traditional tree by introducing more bandwidth into the

switches near the root. A Fat-Tree consists of two sets of elements: the pods, which is a collection of edge and aggregation switches that form a complete bipartite graph, i.e. a Clos graph and the core switches that interconnect the pods. In addition, each pod is connected to all core switches generating a second Clos topology. All switches could be identical to avoid expensive switches with high port density in higher topology levels. The number of available ports per switch is the only parameter that determines the total number of pods and in consequence the total number of required switches as well as connected servers. More specifically, if n is the number of ports on each switch, then there are n pods, with $n/2$ edge switches and $n/2$ aggregation switches in each pod. Each pod is connected with $n^2/4$ core switches and with $n^2/4$ servers. Thus in total, there are $5 * n^2/4$ switches that interconnect $n^3/4$ servers. Fat-Tree allows all servers to communicate at the same time, providing the total capacity of their network interfaces.

2.4.2.4 BCube

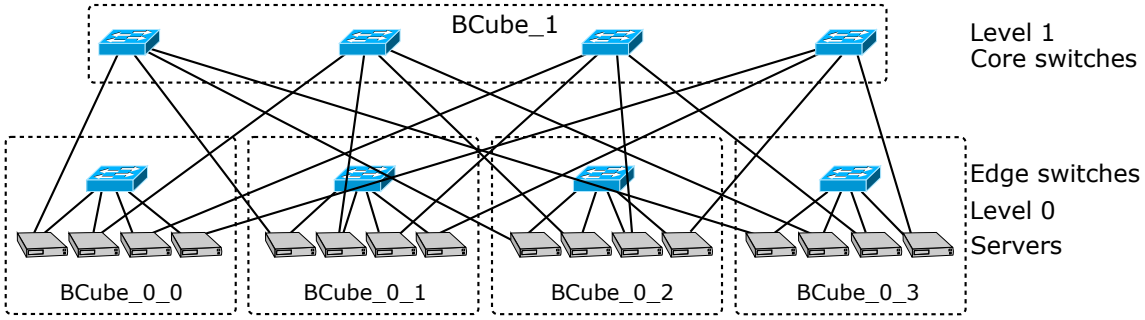

Figure 2.8: BCube architecture

BCube, as shown in Figure 2.8, is a DC architecture with a recursively defined structure [37]. The main difference to the other architectures described above is that the servers are part of the network forwarding infrastructure, i.e. they forward packets on behalf of other servers. The main module of a BCube topology is $BCube_0$, which consists of n servers that are interconnected by an n-port switch. A $BCube_1$ is constructed using n $BCube_0$ networks and n switches. Each switch is connected to all $BCube_0$ networks via its connection with one server of $BCube_0$. More generally, a $BCube_k$ ($k \geq 1$) consists of n $BCube_{k-1}$s connected by n^k n-port switches. The number of connected servers as well as the number of required switches in a BCube is a function of n, the total port number of each switch, and k, the number of BCube levels. Compared with Fat-Tree, BCube provides better one-to-x server support, i.e. higher network capacity for one-to-x (one-to-one, one-to-many and one-to-all) traffic between servers.

2.4.2.5 DCell

DCell [38] is defined recursively and uses servers and switches for packet forwarding. The main module is $DCell_0$, which is composed of a switch connected to n servers. A $DCell_1$ (Figure 2.9) is built by connecting $n+1$ $DCell_0$ networks. $DCell_0$ is connected to all other $DCell_0$ cells via one link from one of its servers to a server in another $DCell_0$. Differently from BCube, the switches are connected only to servers within the same DCell and the connection between

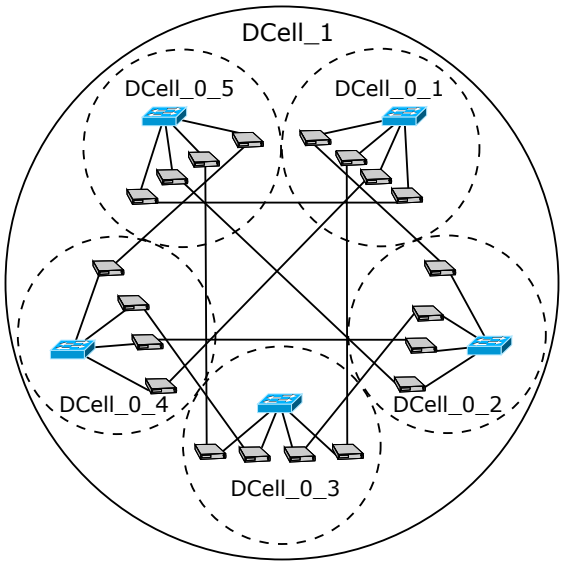

Figure 2.9: DCell architecture

different DCell networks is always done by using servers. Though the degree of DCell is small, the high-level links in a DCell may travel a relatively long distance. The disadvantages of DCell are higher wiring cost and more and longer communication links.

DCell and BCube have more node-disjoint paths and link-disjoint paths because of the multiple ports on their servers, which results in higher redundancy level. Unless each of the node-disjoint paths has a failure on it, two servers will always remain connected. Further it should be noted that the redundancy level of DCell or BCube is decided by the number of ports on servers.

2.4.2.6 Summary: Challenges

Scalability is one of the challenges to the DC networks. With cloud computing, network virtualization and NFV, DCs are required to scale up to hundreds of thousands of nodes. Furthermore, the DC networks are also required to deliver high cross-section bandwidth. However, traditional DC network architectures, such as three-tier DC network offer poor cross-section bandwidth and have high oversubscription ratio near the root. Fat-Tree architecture delivers 1:1 oversubscription ratio and high cross section bandwidth. DCell offers immense scalability, but it delivers very poor performance under heavy network load and one-to-many traffic patterns [34, 33, 38].

The described DC topologies are later used in the trade-off study in Chapter 4 to compare the suitability for NFV type applications.

2.5 Failure Analysis of Different Networks and their Components

In this section a failure analysis is given for IP networks in general and also fiber transport networks and DC networks. The different types and characteristics of failures in these networks are described.

Failures can occur at different layers in the network. For example at the physical layer a fiber cut may cause a physical disconnectivity. Hardware failures, router processor overloads, software errors, protocol implementation and misconfiguration errors may lead to a disconnectivity between routers.

2.5.1 Types and Characteristics of Link Failures in IP Backbone Networks

First, the link failures in IP backbone networks in general are presented. IP link failures occur due to several causally unrelated events at or below the IP layer. The loss of connectivity between two routers is referred to as a link failure. When network components (such as routers, linecards, or optical fibers) have multiple IP links in common their failures affect all the links [39]. In [40], [39] and [41] the characteristics and types of (network-wide) link failures in the Internet backbone have been analyzed with the use of failure logs over several months.

2.5.1.1 Duration of link failures

Link failures occur as part of an everyday operation and the majority of them are short-lived (less than 10 minutes). Further, only 10% of failures last longer than 20 minutes. These are possibly caused by fiber cuts and/or equipment failures/upgrades. About 40% of the failures last between one minute and 20 minutes. These are possibly caused by router reboots, software problems, transient equipment problems, short maintenance operations on equipment or optical fiber etc. About 50% of all failure events last less than a minute [40].

The Internet backbone infrastructure exhibits significantly less availability and a lower MTTF than the Public Switched Telephone Network (PSTN) [41]. The majority of Internet backbone paths exhibit a MTTF of 25 days or less, and a MTTR of 20 minutes or less. Internet backbones are rerouted (either due to failure or policy changes) on the average of once every three days or less. Routing instability inside of an autonomous network does not exhibit the same daily and weekly cyclic trends as previously reported for routing between inter-provider backbones, suggesting that most inter-provider path failures stem from congestion collapse. A small fraction of network paths in the Internet contributes disproportionately to the number of long-term outages and backbone unavailability [41]. The analysis of the failures in the inter-domain paths between providers shows that 40 % of failures are repaired in under ten minutes. The majority (60 %) are resolved within 30 minutes [41]. Backbone links would be more reliable than access links. The explanation could be that backbone links affect more customers, thus being better maintained and monitored than access links [42]. The overall uptime for all backbone routers averaged above 99.0 % for one year [41].

2.5.1.2 Classification of failures

Failures can be classified based on the causes such as maintenance activities, router-related and optical layer problems.

First, failures can be separated due to scheduled **maintenance** from **unplanned** failures.
Failures resulting from scheduled maintenance are unavoidable in any network. Maintenance is usually scheduled during periods of low network usage in order to minimize the impact on performance. Results indicate that about 20% of all failures is due to planned maintenance activities [39, 41]. Most failure events are due to software upgrades with hardware upgrades the next most frequent cause [42]. While hardware-related events cause the most downtime, software upgrades are responsible for much less of the total downtime. While most of the failure events are scheduled, most part of downtime can be attributed to unexpected failures-likely as planned downtime is as limited as possible. The median planned outage lasts less than 5 minutes. Often network operators are not notified by external entities ahead of incidents that impact the network's operation [42].

Further, the failures can be distinguished between shared link failures and individual link failures, depending on whether only one or multiple links fail at the same time. Almost 30% of the unplanned failures are shared by multiple links and can be attributed to[39] router-related and optical equipment-related problems. The other 70% are single link failures.

Shared link failures indicate that the involved links share a network component that fails. This component can be located either on a common router (e.g. a linecard or route processor in the router) or in the underlying optical infrastructure (a common fiber or optical equipment) [43, 39]. The category of shared failures can be divided into simultaneous and overlapping failures [39, 43].
Simultaneous failures start and/or finish at exactly the same time with an accuracy of microseconds. Simultaneous failures are likely due to a common cause (e.g. sharing of a common component which fails and causes all links at the same time). From the analysis all involved links of a simultaneous failure event were connected to a common router and there was no simultaneous failure event that did not involve a common router. Therefore, these events and failures can then be called router events/router-related [39]. Such problems include a router crash or reboot, a linecard failure or reset, or a CPU overload. However, in 50% of the router events identified in the data set all links came up at exactly the same time; in 90% of the cases the last link came up no later than 2 minutes after the first link. Router-related events are responsible for 16.5% of unplanned failures. They happen on 21% of all routers. 87% of these router events happen on backbone routers and the remaining ones happen on access routers.
Overlapping failures on multiple links can happen when these links share a network component that fails and the listener records the beginning and the end of the failures with some delays. For example, a fiber cut leads to the failure of all IP links over the fiber but may lead to overlapping failures for several reasons. It turns out that 80% of all overlapping failures in the study do not share a common router and can be considered to be optical-related since they share a number of underlying optical component that fails (e.g. a fiber or another optical equipment). Shared optical-related failures are responsible for 11.4% of all unplanned failures. Short time-to-repair values are more likely due to faults in the optical switches, while longer times correspond to fiber cuts or other failures that require human intervention to be repaired.

Individual link failures are those that affect only one link at a time. These links are highly heterogeneous, which means a number of links fail significantly more often than others. Individual failures can be divided into high failure and low failure links depending on the number of failures per link. High failure links include only 2.5% of all links; however, they are respon-

sible for more than half of the individual failures. Further, they are responsible for 38.5% of all unplanned failures, which is the largest contribution among all classes [43, 39].

2.5.2 Types and Characteristics of Failures in Fiber Networks

Next, optical-related networks, especially fiber networks are presented. Fiber cuts are the most dominant failures in today's telecommunication networks.
A fiber cut usually occurs due to a duct cut during construction or destructive natural events, such as earthquakes etc. All the light paths that traverse the failed fiber will be disrupted, so a fiber cut can lead to tremendous traffic loss. Other optical network equipment, like OXC, amplifier etc. may also fail [32].

Different optical cable installation methods exist:
The installation of optical cables can be done with the trenchless technique, mini-trench technique, micro-trench technique. The cables can be installed in underground ducts, in tunnels and on bridges, along railways or in sewer ducts. Further installation of buried cables, marinized and submarine optical cables, aerial cables or optical ground wire (OPGW) cable are common. OPGW cable technology is specifically designed for high voltage power line installations. OPGW has the advantage of using the ground wire of a power line for communications, too.

Since optical cables are installed in various environments (aerial, buried, duct, tunnel, underwater etc.) they are exposed to different environmental conditions. The range of environmental conditions must be considered to determine the cable construction that will continuously maintain the desired characteristics. The external factors relating to the various environmental conditions can be divided into two categories: natural external factors (temperature, wind, water, earthquakes etc.) and man-made factors (smoke, air pollution, fire etc.) [30].

2.5.2.1 Causes of Fiber-Optic Cable Failures

Causes of cable breaks in buried fiber-optic cables: Cable breaks in direct buried cables are mostly due to excavation (80%), rodents (5%), workmen, flood, lightning. The failures can occur along public right-of-way such as roadways and utility easements, and private right-of-way such as railroads and pipelines. Cable breaks of cables installed in ducts are due to excavation (65%), workmen (13%), rodents, extreme temperatures (e.g. water frozen inside a splice closure). Immediate causes of sub-surface cable failures are dig-ups (71%), process error (failures caused by telco personnel performing maintenance or installation work), rodent, sabotage, flood, vehicles [44].

Excavation (Dig-ups): An excavation failure is defined as damage to fiber-optic cable during an attempt to penetrate the ground. Most of the excavation failures (71%) of the fiber-optic cables were accidentally performed by responsible companies such as electric and gas, water and sewer, telephone, and highway and road [45]. In most cases (86%) the excavator involved in the dig-up was not working on behalf of the cable owner, while 14% of the dig-ups were caused by the telephone company or their contractor [45]. Examining the cause showed that 33% of reported dig-ups resulted from the excavators' failure to notify the facility owner before digging started. However, 40% of the reported dig-ups occurred in spite of prior notification by

the excavator, accurate cable location, and proper temporary marking of the sub surface cable route [44].

Causes of cable breaks in aerial fiber-optic cables: Immediate causes of aerial cable failures mostly are vehicles damage (i.e. vehicle collisions with utility poles which support aerial cable) (34%), power line contact (24%), fire, firearm and falling trees [44]. Failures in OP-GW cables are mostly due to lightning (25%), installation defect (19%), firearms/hunters, high winds (e.g. tornadoes and hurricanes) [45].

Comparison buried versus aerial fiber: Conventional buried cables are significantly more sensitive to human-related damage than aerial cables. With an eight-hour-average restoration time for buried cables, buried fiber-optic systems experience significantly greater lost revenue and greater maintenance cost than aerial cables. Excavations have no impact on aerial fiber-optic cables [45]. There is no significant difference in the relative failure probability between sub-surface and aerial fiber cables and between direct buried cables and cables installed in underground ducts [44].

2.5.2.2 Calculation of Optical Fiber Cable MTBF Value

For optical transmission links it is common to consider the Cable-Cuts (CC), indicating the average cable length (in kilometers) that has one cable-cut per year. The total cable length is the actual length of the link from one switch/node to another. The calculation of optical fiber cable MTBF value is commonly calculated [29] by considering measured values for the average cable length CC. The MTBF value (in hours) of optical fiber cable, from which the availability can be derived using the MTTR, can be calculated as:

$$MTBF\ (hours) = \frac{CC\ (km) \times 365 \times 24}{total\ cable\ length\ (km)} \tag{2.4}$$

2.5.2.3 Example Values for Availability and MTBF for Fiber Components

For this thesis it is not only important to know how, when and where the failures occur. However, the expected reliability and availability values in the fiber network and its components do matter, too. As for this thesis the wide area transport network scenario is considered, the availability of each individual optical fiber cable also needs to be determined. Therefore, for the most important components the reliability values (as MTBF, MTTR or FIT values) and availability are obtained.

The following tables show these values for fiber cable (Table 2.2) and components (Table 2.3) used in optical fiber networks. For a number of the components three sets of reliability data are provided: optimistic = best-case scenario, nominal = average of values taken from different information sources and conservative = pessimistic estimation of reliability. For values with no further specification on the reliability, are considered as nominal.

Table 2.2: MTBF and MTTR values from different papers for fiber cable

Component [Reference]	MTBF (hours) for 1 km	CC (km)	FIT (FIT/km)	MTTR (hours)
Fiber, aerial [29]	1.75×10^5	20		6
Fiber, buried (optimistic] [29]	5.5×10^6	628		9
Fiber, buried (nominal) [29]	2.63×10^6	300		12
Fiber, buried (conservative) [29]	2.41×10^6	275		24
Fiber, submarine (nominal) [29]	4.64×10^7	5300		540
Optical fiber (PON) [46]	5×10^6		200	14
Fiber [47]	1.75×10^6		570	24
Fiber and inline amplifier [48]	3.23×10^6		310	12
Fiber [49]	8.773×10^6			-
Fiber [49]	4.99×10^6			-
Fiber [49]	3.21×10^6			-

Table 2.3: MTBF and MTTR values for components in WDM network

Component [Reference]	MTBF (hours)	MTTR (hours)
Transponder [50]	1.96×10^5	2
Transponder 2.5Gbps (optimistic) [29]	5.00×10^5	2
Transponder 2.5Gbps (nominal) [29]	4.00×10^5	6
Transponder 2.5Gbps (conservative) [29]	2.94×10^5	9
Transponder 10Gbps (optimistic) [29]	9.6×10^5	2
Transponder 10Gbps (nominal) [29]	3.5×10^5	6
Transponder 10Gbps (conservative) [29]	2.94×10^5	9
Multiplexer [50]	6.06×10^5	2
bidir Multiplexer-demultiplexer (optimistic) [29]	10.00×10^5	2
bidir Multiplexer-demultiplexer (nominal) [29]	1.67×10^5	6
bidir Multiplexer-demultiplexer (conservative) [29]	1.00×10^5	9
Demultiplexer [50]	2.79×10^5	2
Booster amplifier [50]	2.11×10^5	2
Pre-amplifier [50]	3.70×10^5	2
Amplifier receiver [50]	2.10×10^5	2
Amplifier terrestrial [50]	2.11×10^5	2
Amplifier submarine [50]	20×10^6	336
bidirectional Optical Amplifier (optimistic) [29]	5.00×10^5	2
bidir Optical Amplifier (nominal) [29]	2.50×10^5	6
bidir Optical Amplifier (conservative) [29]	1.00×10^5	9
Optical cross connect (OXC) (nominal) [29]	1.00×10^5	6
OXC redundant 1+1 protection (nominal) [29]	2.06×10^5	4

Table 2.2 shows MTBF and MTTR values for fiber optical links taken from the literature for fiber cable. It can be seen that realistic MTBF values range between 1.5 and 5.5×10^6 hours for 1 km of fiber. Since fiber-cuts have to be located and eventually excavated MTTR values in the range of several hours are required [44]. The mean time to completely repair a failed cable is slightly over 14 hours while the mean time to restore service is slightly over five hours [44].

2.5.3 Types and Characteristics of Failures in Data-Center Networks

In this part the types and characteristics of failures in DC networks are examined. The most important devices in a DC are the servers and switches, which are the main devices to be examined here.

DC failures can be hardware failures like switch (core/aggregation) failures or rack/ToR switch failure, server failures or link/connection failures between switches and/or servers.

2.5.3.1 Failure Behavior of the Network

Intra-DC and inter-DC network problems are the reason for network failures which cause significant impact to cloud services. This impact is dominated by connectivity loss problems (70%) and service errors (43%) due to intra-DC and inter-DC network problems, respectively [51].

Inter-DC: For inter-DC network link failures, link flapping (e.g. due to routing protocol issues and convergence) dominate the problem root causes (36%). Depending on the protocol timers, such an event is observed as a 'link flap', yet the true underlying cause could be an optical re-route, possibly in response to a fiber cut. The second major root cause is high link utilization (29%) followed by unplanned changes (6%) [51]. Fiber length in inter-DC links has no statistical correlation with the number of failures. Links with high utilization exhibit 2 - 3 times higher downtime than expected [51]. Inter-DC links have the lowest failure rate (fewer than 3% of each of these link types failing in one year). However, inter-DC links take the longest to repair [52].

Intra-DC: For intra-DC network failures there is a broad range of problems such as hardware faults (e.g. device failures, memory errors), operating system bugs and misconfigurations. The Intra-DC network failures are dominated by connectivity errors (64%-78%), hardware failures (20%-73%) and software problems (7%-24%) across Layer-3 and Layer-2 devices. Most failures are short-lived occurring when the device unexpectedly reloads. Interface-level errors, network card problems, and unexpected reloads are notable for all device types in DC. Interface errors usually last for about 5-7 minutes. While the service would be still available, its users may experience high latency or packet drops [51].
For network-related failures in DC link failures happen about ten times more than node failures per day. Usually node failures are due to maintenance [52].
Management links and inter-DC links have the lowest failure rate. ToR switches are most reliable with the lowest mean number of failures due to their large population. However, ToR switches have the most downtime as they have a low priority for repair compared to other components [52]. Layer-2 aggregation switches exhibit high availability when about half of their port capacity is utilized in terms of ToR switch count. However, the availability significantly

decreases as the ToR switch count reaches the full switch capacity. Therefore, to deliver highly reliable and cost-effective services, scale-out switches (i.e. small port-count, low-cost commodity switches) with low to medium port density may deliver higher availability in comparison to their expensive higher capacity equivalents [51].
Load balancers (LBs) are least reliable and experience many short-lived faults. Root causes of failure for LB are mainly the software bugs, configuration errors and hardware faults. The links forwarding traffic from LBs have the highest failure rates [52].
Device failures are not memory-less, i.e. they are not independent [51]. After one failure the probability of subsequent failures is higher in the near time window. However, the probability of multiple failures is quite low ($< 0.05\%$). When devices fail multiple times, this happens mostly within one week of repairing due to ineffective repairs and problem misdiagnosis [51]. Repairs are relatively more effective for access routers and aggregation switches. ToR switches exhibit an increase in the probability of device failure after repair indicating that their repairs are not very effective [51].
Network redundancy reduces the median impact of failures (in terms of number of lost bytes) by only 40%. The overall DC network availability is about 99.99% for 80% of the links (between switches) and 60% of the devices (core, aggregation, ToR switches) [52]. Network redundancy is least effective at the access router-aggregation switch layer and is most effective at the inter-DC level [51].

2.5.3.2 Failure Behavior in Servers

For 92% of the physical machines (PM)/servers no repair events occur. However, the average number of repairs per year for the remaining 8% is 2 per machine. About 78% of total faults/replacements were detected on hard disks, 5% on RAID controllers and 3% due to memory failures. About 13% of replacements were due to a collection of components (not particularly dominated by a single component failure) [53]. Hard disks are clearly the most failure-prone hardware components and the most significant reason behind server failures. About 5% of servers experience a disk failure in less than one year from the date is commissioned (young servers), 12% when the machines are one year old, and 25% of the servers experience hard disk failures when they are 2 years old. None of the following factors like age of the server, its configuration, location within the rack and workload run on the machine were found to be a significant indicator for failures [53]. In [54] they found out that annual disk replacement rates typically exceed 1%, with 2-4% common and up to 13% observed on a number of systems. This suggests that field replacement is a fairly different process than one might predict based on data sheet MTTF. Based on records of disk replacements, failure rate is not constant with age and that, rather than a significant infant mortality effect, they see a significant early onset of wear-out degradation. That is, replacement rates in their data grew constantly with age, an effect often assumed not to set in until after a nominal lifetime of 5 years [54]. The major failure reasons of PM failure are server components (disk and processor), wear-and-tear of server, over-aggressive consolidation/repeated on-off cycles and temperature rise. The repeated on-off cycles can decrease the lifetime of a PM [55].

In [56] a number of differences in PM and VM failures are presented. VMs have lower failure rates and lower recurrent failure probabilities than PMs. Inter-failure times of VMs show a similar behavior as for PMs. Software inter-failure times are the shortest, compared with

hardware/infrastructure-related ones. The average repair time of VM failures is lower than for PM by almost a factor of two since hardware failures take longest to repair. The relationship between VM failures and VM age does not follow a bathtub-like function. VM failures show a weak positive trend with age. Therefore, periodically taking snapshots of existing VM images and creating new VM instances may reduce VM failures. Resource utilization is more critical than capacities for PM failures, in particular, CPU utilization. The higher the CPU usage the higher the temperature rise and, therefore, reduces the server lifetime [56]. Each 10 degree C temperature rise would reduce the server component life by 50% [55]. The key resource attributes affecting VM failures are the CPU utilization and the number of disks (disk capacity has the least impact) [56]. The majority of PMs show increasing failure rates with respect to CPU utilization, while most VMs show a decreasing trend. The VM failure rates decrease with the consolidation level [56]. Therefore, lower failure rates can be achieved when a fair number of underutilized VMs are consolidated in the systems. There seems to be no impact on VM failures when frequently turning on/off of the VMs [56].

2.5.3.3 Example Values for Availability and MTBF for Data-center Network Components

For the study in this thesis, also reliability and availability values for the DC network components are needed. Several data are collected from references [51], [52], Cisco switches [57] and Intel servers [58] data sheets. The MTBF values of the servers span quite a large range due to the reason that low-cost servers and high expensive servers are considered here. The collected data for MTBF and MTTR values are summed up in the following Table 2.4.

Table 2.4: MTBF and MTTR values for different data-center components

DC component	MTBF (hours)	MTTR (hours)
Server	$0.6667 \times 10^4 - 10.95 \times 10^4$	7-8
ToR switch	$14.5 \times 10^4 - 17.52 \times 10^4$	2.9
Aggregation switch	$8.76 \times 10^4 - 20 \times 10^4$	2.1
Core switch	60×10^4	2.1

For the further chapters, the MTBF and MTTR values from Table 2.4 are used to determine the availability values for each component.

2.6 Virtual Network Embedding

2.6.1 General Virtual Network Embedding Problem

The Virtual Network Embedding (VNE) problem (sometimes also called slice embedding problem) deals with the efficient mapping of a set of Virtual Network Requests (VNRs) to physical nodes and links. A VNR is a set of virtual nodes that must be mapped to a set of physical nodes with sufficient resources to accomplish the requirements and a set of virtual links to be mapped to a set of paths in the physical network. This embedding is supposed to optimize the

allocation of physical resources. The embedding can be optimized with regard to performance (e.g. CPU capacity, link bandwidth), energy-efficiency (e.g. power usage of a node), security (e.g. node reliability, link encryption) or other parameters. Furthermore, a newer embedding must not interfere with the operations of previously existing VNs.

Mapping a virtual link to a path in the physical network obviously uses resources of the physical links on the path. Furthermore, several virtual links can use the same physical link (i.e. a physical resource can be partitioned into several virtual resources). For the mapping of virtual nodes and virtual links, several algorithms can be considered. Due to the differences in node and link mapping, most algorithms treat node and link mapping as separate steps. Due to the combination of node and link constraints, the resulting VNE problem is NP-hard. Practical algorithms, therefore, use heuristic approaches to obtain a near optimum value [59, 60, 61].

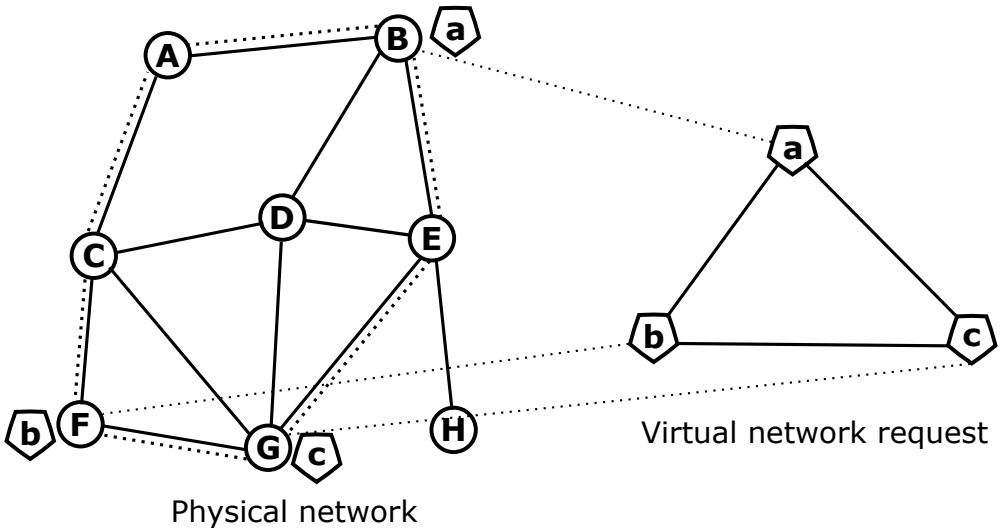

Figure 2.10: Virtual network embedding with physical network graph $G_s = (V_s, E_s)$ and virtual network request graph $G_v = (V_v, E_v)$

Formal Problem Description:

The physical network (also called substrate network) is presented as a graph $G_s = (V_s, E_s)$ where vertices V_s represent the physical nodes and edges E_s represent the links between nodes in the network (see Figure 2.10). Both physical nodes and links have constraints. Node constraints can be CPU, RAM, geographical location etc. Link constraints can be bandwidth, delay etc.

The virtual network request (VNR) consists of virtual nodes and virtual links, which is also described by a graph $G_v = (V_v, E_v)$ with constraints that describe the requirements of the virtual nodes and virtual links (see Figure 2.10). The mapping of virtual nodes and links onto the physical network is realized by an embedding algorithm.

The objective of the VNE is to find an effective and efficient embedding algorithm for the VNR. Embedding has been proven to belong to the NP-hard category of problems in [62, 63].

There exist different types of approaches to solving the VNE problem. One approach is to find an exact solution for the VNE problem. However, only small instances of the problem

can be solved optimally due to the NP-hard category of the VNE problem. Using an exact approach can create baseline solutions that represent an optimal bound for heuristic-based VNE solutions. Linear Programming (LP), especially Integer Linear Programming (ILP) or Mixed Integer Linear Programming (MILP)/Mixed Integer Programming (MIP) can be used to formulate the VNE problem as an optimization problem [64]. Another approach is to solve the problem using a heuristic. With a heuristic a sub-optimal, however, acceptable solution can be found within a short time. Three approaches are commonly used to solve a heuristic for the embedding problem: (brute-force) backtracking [65], simulated annealing [66] and approximation algorithms [67].

Backtracking incrementally builds candidates to the solutions and abandons each partial candidate c ('backtracks') as soon as it determines that c cannot possibly be completed to a valid solution. Simulated annealing is a heuristic optimization solution. Each point s of the search space is analogous to a state of some physical system and the function to be minimized is analogous to the internal energy of the system in that state. The goal is to bring the system from an arbitrary initial state to a state with the minimum possible energy. In approximation algorithms it is tried to make the local optimal choice at each stage with the hope of finding a global optimum. Several techniques exist to design approximation algorithms like greedy algorithms, local search or dynamic programming.

A VNE for a VNR is defined as a mapping $\mathscr{M}$ from G_v to a subset of G_s, so that the constraints in G_v are satisfied [62]:

$$\mathscr{M} : G_v \mapsto (V_s, P_s) \tag{2.5}$$

The VNE problem can be divided into two separate problems: virtual node mapping and virtual link mapping.

a) Node mapping:

$$\mathscr{M}_{node} : V_v \mapsto V_s \tag{2.6}$$

The virtual nodes are mapped to resource (physical) nodes in the physical network. One virtual node needs to be mapped to exactly one physical node which satisfies the resource requirements of the virtual node (Equation (2.6)). The node mapping problem is still an NP-hard problem, similar to the multi-way separator problem [62, 63]. For node mapping, greedy methods [62, 68] are often used.

b) Link mapping:

$$\mathscr{M}_{link} : E_v \mapsto P_s \tag{2.7}$$

P_s is denoted as the set of all loop-free paths of the physical network. In the link mapping the feasible paths between all physical nodes mapped from the virtual nodes are established. A virtual link between two virtual nodes can be mapped on a physical path, which could consist of one or multiple physical links (Equation (2.7)). For this problem (k-)shortest path [68] or multi-path algorithms using multi-commodity flow algorithms [69] are used. In (k-)shortest path algorithms each virtual link is mapped to the shortest-path or k number of shortest paths. To find the paths one can use shortest path algorithms such as Dijkstra's algorithm or Bellman Ford algorithm. For k-shortest path the algorithm (e.g. Eppstein [70]) not only finds the shortest path, but also $k-1$ other paths in order of increasing cost with k being the number of shortest paths to find. In multi-path algorithms (using multi-commodity flow algorithms) the mapping

of one virtual link to multiple physical paths is done with different splitting ratio. The multi-commodity flow problem is a network flow problem with multiple commodities (flow demands) between different source and sink nodes.

Figure 2.10 shows a mapping (dotted lines) of a virtual network request (right graph) onto a physical network (left graph).

Existing solutions of algorithms for embedding VNs can be divided into categories: offline and online version of the problem and static and dynamic version.

In the offline version all the VNRs which need to be embedded are known from the beginning. Therefore, results closer to an optimal embedding for the VNRs can be achieved. In the online version VNRs arrive dynamically at different times and need to be mapped in real time. Further the VNRs are not known in advance [59, 60].

Further, the VNE problem can be solved in a static or dynamic version: In the static version (static mapping) there are no changes in resource assignment after the mapping and physical resources allocated are not changeable during the lifetime of a VN [68]. The dynamic (adaptive) mapping allows changes depending on the demand and performance of the VN and resources allocated to a VN can be adjusted on the basis of traffic load to improve the overall network performance. This re-optimizing requires monitoring of the VN and dynamic updates of physical node and link capacities [68].

2.6.2 Survivable Virtual Network Embedding Problem

In this section the focus is on survivable VNE problem and solutions. Survivable or resilient VNE deals with failures in the physical and virtual network. Since multiple VNs can share the physical resources of the underlying physical network even a single failure in the physical network can affect a large number of VNs and the services they offer. Thus, the problem of efficiently mapping a VN to a physical while guaranteeing the VN's survivability in the event of failures in the physical becomes important. The challenges in resilient VNE to be considered are link and node failures, which have to be backed up before the failure or recovered after the failure. The survivable VNE problem is a resource allocation problem similar to VNE problem. The difference is that in the resilient VNE interruptions in the network (e.g. communication between the nodes) caused by defect parts/components of the underlying physical network (i.e. physical links and/or physical nodes (e.g. switches, router, servers ...)) also have to be considered.

There are two main survivability techniques: **protection** and **restoration** [71, 32, 72].
Failure protection is done in a proactive way to pre-compute and reserve the backup resources (backup links or backup paths or backup nodes) in advance before any failure happens. The backup resources are created as copies of the original resources (denoted as primary resources). Protection schemes have faster recovery time compared to restoration schemes and can guarantee recovery from service disruptions against which they are designed to protect. However, a disadvantage is that the use of backup resources may result in high cost for reserving resources, especially if no disruption occurs. Reactive mechanisms, which are called restoration mechanisms, react after the failure occurs and start the backup restoring mechanism. Dynamic

restoration schemes are more efficient in utilizing network capacity/resources as they do not allocate spare capacity/resources in advance and provide resilience against different kinds of failures (including multiple failures). However, some data loss is possible in the reactive case.

There exist two kinds of backups for the protection scheme [71, 32, 72]: **dedicated backup** or **shared backup**. In shared backup the resources for the backup may be shared with other backups. In M:N protection M backups are used to protect N primary elements. In the dedicated case the backup resources are not shared for other backups and are exclusively reserved. However, this is resource-inefficient compared to the shared backup.

Failures in the VN can be repaired through re-instantiation of the failed VN element (link or node) on the same physical elements or some other suitable physical elements. Failures in the physical network require more effort to be restored or backed up since sharing the physical can affect several virtual resources.

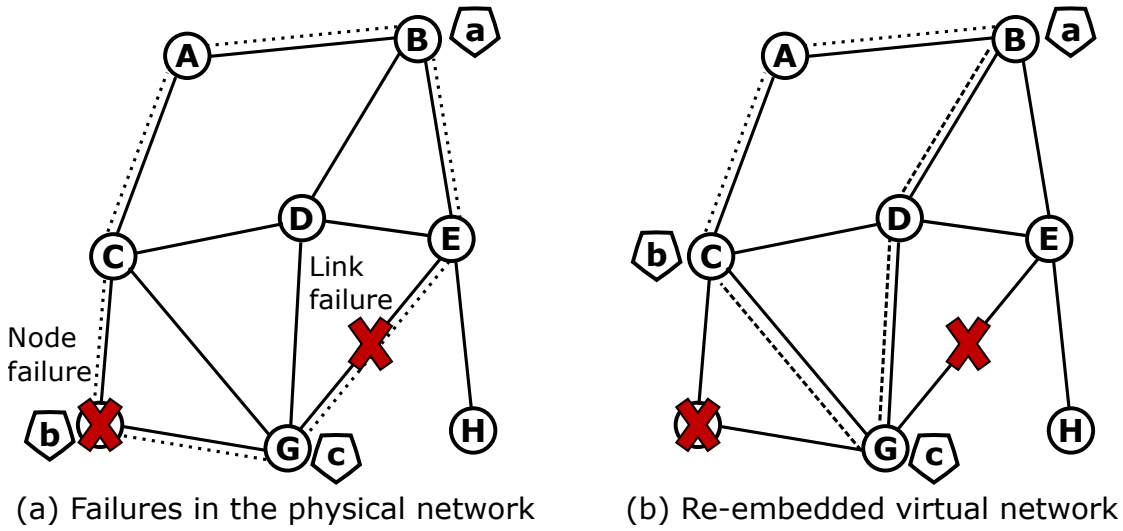

Figure 2.11: Survivable virtual network embedding

After embedding a VNR a physical node and link failure (represented by crosses) occurs as shown in Figure 2.11 (a). The failed node has mapped the virtual node **b**, which needs to be remapped. The physical link failure is on the physical path for the virtual nodes **a** and **c**. A possible re-embedding of the virtual network on the physical network after the failure is drawn in Figure 2.11 (b), where virtual node is migrated to a new physical node and the links are re-embedded for the migrated node and the failed physical link.

For a physical node failure alternative physical nodes have to be found and the affected virtual node or nodes including the affected links have to be migrated. For a physical link failure a backup path over different physical links has to be found, which can be achieved using a path or link-based method. For the path-based method each E2E primary path is backed up by a disjoint path from the source node to the destination node. Link-based method means that each primary link is backed up by a pre-configured bypass path.

Figure 2.12 (a) shows the path-based method (path protection) for the backup of the link failure between physical nodes **E** and **G**. The complete path from node **B** to node **G** is protected with

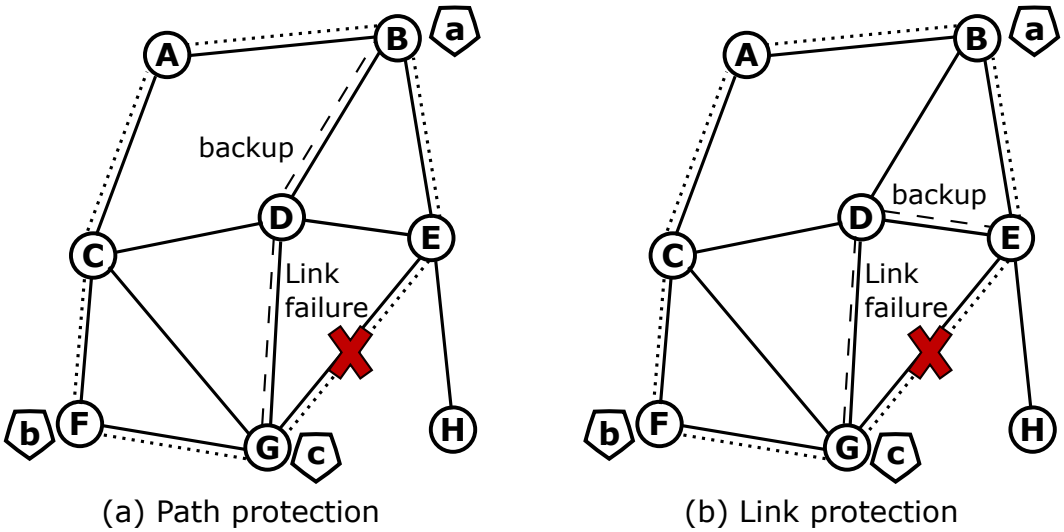

Figure 2.12: Protection methods for physical link failures

this method while with the link-based method (link protection) as shown in Figure 2.12 (b) only the link between node **E** and **G** is protected.

The task is to embed a VN that can deal with virtual and physical network failures in a way that after the failure the VN is still operating. The failure and the fixing/recovery should be transparent to the users of the VN.
One possibility can be to extend the VN graph with backup nodes V^B and backup links E^B (Equation (2.8)) and embed the extended graph G_v^B. The backup links E^B are links between backup nodes and working nodes.

$$G_v^B = (V_v \cup V^B, E_v \cup E^B) \tag{2.8}$$

In the survivable mapping virtual nodes of one VN should not be mapped on the same physical node due to the fact that a possible failure of this physical node could affect several virtual nodes. For links, different virtual links should use distinct paths in the physical network.

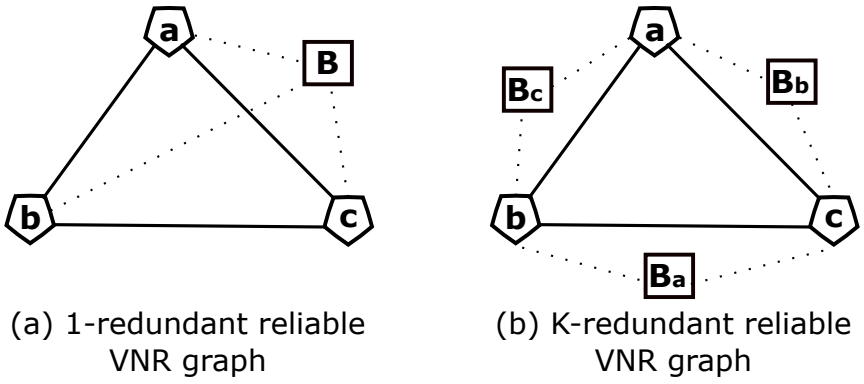

Figure 2.13: VNR graph with backup nodes

Figure 2.13 (a) shows a VNR graph with one backup node *B*. This backup node is shared by each of the three virtual nodes **a**, **b**, **c** as a backup if one of the primary/original nodes would fail. (b) shows a VNR graph with each virtual node having a backup node (dedicated backup). For reliable VNE the backup graphs created need to be mapped to the physical network.

2.6.3 Algorithms for the Virtual Network Embedding

2.6.3.1 Algorithms for the General Virtual Network Embedding

The topic of the VNE problem and its algorithms have been well studied in recent years. Many different basic solutions for embedding VNs exist in the literature. A detailed survey about VNE algorithm and classification can be found in [60].

Since the VNE problem is NP-hard, it is attempted to reduce the problem space by the limitation of the problem. However, this VNE problem with reduced complexity still remains NP-hard. Restrictions are:

1. Considering the offline version of the embedding problem (i.e. all the VNRs are known in advance and the VNRs can be ordered in advance) [68, 73].

2. Ignoring either node requirements or/and link requirements (e.g. ignore processing resources on the nodes or bandwidth on links [73], both links and nodes were unconstrained [68]).

3. Assuming physical network resources are unlimited (i.e. infinite capacity of resources of the physical nodes and links like unlimited bandwidth or CPU) to avoid admission control (i.e. reject or postpone a number of VNRs to meet the resource guarantees for existing VNs) [68, 73].

4. Focusing on specific VN topologies (e.g. star topology, tree topology) [73].

5. Assuming complete knowledge of the physical network, although in reality PIPs keep the topology information and traffic matrices secret [73].

6. Focusing on single PIP environment, however, in reality heterogeneous domains belong to different PIPs [73, 62].

7. Assuming homogeneity in PIP environment, however, for E2E embedding domains are heterogeneous with different granularity in the resources (e.g. different link connectivity).

8. Assuming the physical network after embedding be operational at all times (ignoring possibility of physical link or/and node failures), not considering the survivable VNE.

Various work simplifies the problem by decoupling the node and link embedding process, such as in [68, 73, 62]. On the other hand, approaches are found to model or design heuristics to coordinate the node mapping and link mapping in one stage, as in [65, 67].

Path splitting and path migration can be used to enable better resource utilization by allowing the substrate to accept more VN requests as in [62]. Path splitting divides the available bandwidth of the path into small bandwidth to satisfy the resource constraints. The division of bandwidth over the physical paths is specified as a splitting ratio. Path migration is the changing of the route or the splitting ratio of a virtual link while the node mapping is kept fixed. Although effective, path splitting may cause packet redirection, increased routing table size and packet reordering.

Further, many algorithms assume simple cost functions at minimizing resource consumption (especially bandwidth) or maximizing the revenue and acceptance ratio of VNRs [68, 62, 65, 67]. In [74] a different cost function is used for the VNE, where the cost increase exponentially as the link traffic increases. This cost function is used to measure the cost of a VNE in a physical network. The exponential increase of the cost function forces to find an embedding where the physical paths of the virtual links should be as short as possible. Therefore, the number of embedded VNs could be increased.

Many approaches focus on solving the VNE problem in a single PIP environment. However, there are a few which already consider a multi-domain VN environment [75, 76, 64, 77, 78]. PolyVine [75] is a distributed policy-based protocol that coordinates VNE between PIPs. It ensures competitive prices for service providers. In [76] a hierarchical resource discovery framework with virtual resource description is presented. If the virtual resources are matching from multiple PIPs the VNP coordinates the multiple PIPs to interconnect the virtual edge nodes. Intra-PIP VNE is done by each PIP separately. In [78] the embedding of VNs in a networked cloud environment is studied. A k-cut algorithm to generate inter-domain VN partitioning is used. For those generated candidates the least-cost inter-domain linkages are selected using a heuristic based on Gomory-Hu trees.

2.6.3.2 Algorithms for the Survivable Virtual Network Embedding

This section discusses state-of-the-art algorithms and methods for survivable/resilient VNE for link or node failures.

2.6.3.2.1 Survivable VN Embedding against Link Failures

Link failure survivability problems and survivable routing have already been investigated for optical [71, 50, 79, 48] and multi-protocol label switched (MPLS) networks [80]. However, the problems studied there are offline versions and assume the traffic demand matrix has been available in advance which is often not the case in VNE. Further, in VNE ensuring that all virtual links are intact when failures happened is necessary. In optical networks it is often enough that all nodes remain connected in the presence of failures even if the connection is not via a direct virtual link.

The following algorithms embed VN against links failures in the physical network.

Link restoration and protection methods
A reactive backup mechanism to protect against a single physical link failure for VNE is pro-

posed in [81]. The idea is a fast rerouting of the links and to reserve bandwidth for backups on each physical link. The polynomial time heuristic consists of three parts. Before any VNR arrives backup paths for each physical link are calculated with a path selection algorithm. Then node and link embedding are done for the arriving request with an existing embedding algorithm. When a physical link failure occurs the calculated backup paths are used to reroute the bandwidth of the affected link using a reactive online optimization mechanism. The optimization goal is to maximize revenue for the PIP. This backup mechanism is a restore approach. Therefore, after a failure it cannot guarantee 100 % recovery. In cases that the bandwidth resources are used for new VNRs, there may not be enough resources left for the recovery. Further, with the increase in traffic load, a failure can cause a large amount of data loss and the backup mechanism may not restore the VN.
The problem of shared backup network provisioning for a single physical link failure for VNE with a link-based backup approach similar to [81] is investigated in [82]. Two schemes are proposed: In Shared On-Demand approach, bandwidth resources are allocated to the primary flows and to restoration flows when a new VNR arrives. Bandwidth sharing is possible for the restoration flows, however, not for the primary flows. After every VNE the residual resource information needs to be updated. In Shared Pre-Allocation approach, backup bandwidth for each physical link is pre-allocated during the configuration phase before any VNR arrives. Since the bandwidth pre-allocation only needs to be done once and not for every VNR, there is less computing done during the VNE phase. The overall optimization is to maximize the revenue for the PIP through accepting most VNRs. The advantage to the previous algorithm [81] is that the backup bandwidth is already allocated before the failure happens and not after the failure. The disadvantage of the Shared Pre-Allocation approach is that backup bandwidth is reserved independent of the VNRs and may not be used at any time if few VNRs arrive.

Path protection methods
Instead of backing up each primary link as in [81] and [82] a path protection method for single physical link failure is developed in [83]. To protect a virtual link against failures its primary path (where the virtual link is mapped to) and backup path should not share any common physical link or node. This is known as disjoint paths. However, using disjoint paths results in high expenses of bandwidth usages. Therefore, if possible, bandwidth is shared among backup paths that pass the same physical link to save bandwidth resources. In this approach, only bandwidth constraint is considered.
Through a node migration technique in addition to path protection, the problem of survivability for link failure is tried to be solved by optimizing the networking and computing resources in [84]. Their approach, migratory shared protection, migrates and maps a VN node to another physical node to increase the resource efficiency when a failure occurs. The relocated node should need less backup path length to the destination node than before the migration and save resources. All VN links connected with the migrated VN node have to be remapped and the backup links must be link-disjoint to the primary links. The re-established paths from the new migrated node build a tree: the migratory backup tree. For this protection method, intra-share can be applied, which means sharing resources among the migratory backup tree and the corresponding migrated primary paths. Also, inter-share which means sharing of backup resources between different backup paths is possible. Compared to the traditional backup protection where only one path needs to be migrated, in their approach [84] several links and, at least, one node need to be migrated.
QoSMap is a mechanism attempting to consider both, quality of service (QoS) and resiliency in

constructing VNs over a physical network [85]. Its aim is to map a QoS-specified overlay onto the physical network using direct paths between nodes that are pre-selected possible candidates. Nodes with higher quality are selected first. Node quality depends on the average backup paths that a physical node can provide. Path resiliency is provided by constructing alternate backup paths via one intermediary node that could be additional underlying nodes or selected hosting overlay nodes. However, it may not always be possible to find direct backup paths. Since QoSMap uses direct paths, back-tracking in the algorithm is required to find these (backup) paths. This may take exponential time and affect the scalability of the algorithm.
Since the heuristic QoSMap solution cannot guarantee the best QoS performance, due to sequential and heuristically node selection, this problem is solved with an Integer Linear Program (ILP) in [86] and a heuristic in [87].

Protection through live reconfiguration and migration
Instead of reservation or backup of resources the virtual components can be reallocated/reconfigured when a failure occurs or migrated before the failure occurs. Since this kind of reallocation and reconfiguration approaches do not require backup resources it is a cheaper solution in terms of resources cost. However, the procedure has to be planned in advance and the migrating resources have to be operational all the time to avoid interruption. The authors in [88] present such a strategy based on "opportunistic resilience". In their preventive strategy the bandwidth demand of each virtual link is split over multiple physical paths. As a consequence, physical link failures are less likely to cause a virtual link disconnection. The affected virtual link will remain operational, however, with less bandwidth capacity. Additionally, a reactive strategy is used in order to reallocate the lost capacity over unaffected paths after the failure, trying to restore the bandwidth of degraded virtual links.

2.6.3.2.2 Survivable VN Embedding against Node Failures

The following different approaches try to embed VNs with backup for virtual nodes and protections against node failures in the physical network.

Two-step approaches
In [89] a two-step paradigm to fully recover a VN from facility node failures is presented. A facility node is a physical node with computing capacity. The first step is to construct a graph of the VNR with backup virtual nodes and links and then this enhanced VNR has to be mapped onto the physical network. Two solutions are proposed: the 1-redundant scheme and the K-redundant scheme. A 1-redundant solution is a reliable VN graph with one redundant virtual node (backup node) and redundant connections, which is then mapped onto the physical network. Assuming only single failures the backup node of a certain virtual node can also be used as the backup of a number of other virtual nodes for resource sharing. For the mapping it can share the physical link resources when mapping them onto the physical network (backup share) and also share the bandwidth link resources between the original working path and its associated backup path (cross share). In the K-redundant solution a K-redundant reliable VN graph is designed, in which each critical node is permitted to have a corresponding backup node. The optimization objective is to minimize network resource cost. However, this approach may fail to provide a joint optimization for the allocation of both the active and backup resources. In the worst case there is a need to reserve a backup node for every critical node and link to every

neighbor node.
Another two-step method for surviving single facility node failures is presented in [90]. This approach designs the enhanced VN with a failure-dependent strategy instead of a failure-independent strategy as in the previous one [89]. It manages to further reduce the needed virtual resources and therefore, less allocated backup resources compared to failure-independent strategy. The idea is that when node i fails, the role of node i may be replaced by any other node after a rearrangement of all the nodes (including the backup node(s)) using graph transformation/decomposition and bipartite graph matching. The disadvantage of this approach is that a large amount of possible migrations of working nodes after a failure makes the approach less applicable in large networks.

Node protection with location constraint
The Location-constrained Survivable Network Embedding problem to protect against any single facility node failure is investigated in [91]. The location constraint of a virtual node is considered for its backup node. The goal is to map the VN with minimum resources while satisfying the bandwidth constraints for the links and capacity constraints for the nodes including meeting the location constraints for the primary and protection node. The idea is to construct a graph with the virtual and physical graph in one single graph. Thereby each virtual node is connected to a number of candidate physical nodes, which satisfy the location and capacity constraints. The heuristic algorithm maps first the VNR using an existing embedding algorithm and then the backup request is mapped.

Multi-Failure Protection
In [92] an approach for solving the problem of survivable VN mapping for single regional failures in a federated computing and networking system is presented. A regional failure occurs when a single disruption event causes multiple devise/components in the same regional area to stop operating. In a federated computing and networking system facility nodes from a DC are interconnected. These facility nodes need to be backed up to achieve a survivable VN mapping. Their approach is based on the assumption that the number of distinct regional failures is finite in a specified geographical area and that a regional failure refers to a set of physical nodes and links, which is in the same shared risk group. The proposed approach first solves the non-survivable VN mapping problem with a heuristic and extends this heuristic to handle the survivable VN mapping problem. Two failure dependent survivable VN mapping algorithms are developed. The Separate Optimization with Unconstrained Mapping decomposes the problem into separate non-survivable problems for each possible regional failure plus one for the initial working mapping of the VNR. Each problem is mapped in a way that the cost of the used resources are minimized. The other approach, Incremental Optimization with Constrained Mapping, first embeds the initial working mapping and then handles each regional failure after another. Compared to the Separate Optimization with Unconstrained Mapping the additional computing and networking resources that are needed to handle the failure are tried to be minimized. With this strategy, the mapping of unaffected virtual nodes is not changed. The disadvantage of Separate Optimization with Unconstrained Mapping is the re-calculating virtual mapping of unaffected nodes, which results in more cost and more time to be calculated. Authors in [93] tried to recover from both, physical node and link failures, while minimizing backup resources through pooling. Further a relationship between reliability and the amount of redundant resources is tried to be found. Redundant (backup) virtual servers are created dynamically and are pooled together to be shared between VNs to assure the requested reliability

level. The higher the reliability level, the higher number of backup nodes that are required. It is possible to share the backup nodes in such a way that the total number of backup nodes would decrease as when each VN separately has its own backup nodes. Every backup node can be a standby node for all other critical nodes. With the Opportunistic Redundancy Pooling mechanism backup nodes can be shared between VNs as long as the reliability of every network is satisfied. The Opportunistic Redundancy Pooling shares these redundancies for both independent and cascading types of failures. Therefore, VNs with different reliability guarantees can be pooled together with flexibility in adding or removing VNs to the existing ones.

2.6.3.2.3 Joint Reliability Techniques

A resilient VN design for E2E cloud services is found in [94], which focuses on joint protection of nodes and links. In inter-cloud networks virtual links can connect two different types of nodes: the virtual cloud DC, which hosts services for a particular service provider and the virtual router, which is directly connected to (thus representing) a group of clients. The objective is to protect a service installed in the cloud from becoming unavailable to the clients. Since virtual routers are assumed to be unique geographical locations the approach is aimed at protecting virtual DCs and virtual links. The proposed approach is divided into two different protection strategies: one is offered by the VNP while the other is offered by the PIP. In the first strategy the VNP spreads the services' VMs among two or more DCs from different PIPs. Then it creates disjoint primary and backup paths connecting DCs to virtual routers. To ensure that the protection works properly PIPs are made aware of which virtual links must be disjoint. In the second strategy protection is provided solely by the PIP. The VNP delivers a VNR describing which nodes must remain connected. Then the PIP is responsible for allocating additional geographically different DCs and provide redundant paths for each virtual link.

The authors in [95] propose an approach for VN mapping with combined physical node and multiple logical links protection mechanism. In a first stage that mechanism calculates a cost-efficient VN mapping while minimizing the effects of a single node failure in VN layer. In a second stage a link p-cycle based protection technique that minimize the backup resources while providing a full VN protection scheme against a single physical node failure and a multiple logical links failure is calculated. A p-cycle is a cyclic, pre-calculated, pre-assigned, closed path with a certain amount of allocated spare capacity providing protection for any link that has both end nodes on the cycle. In order to guarantee a full VN protection VN mapping is done in such a way that virtual links belonging to the same VN are mapped to link-disjoint physical paths and node-disjoint physical paths in order to guarantee a node failure independent path protection scheme.

The work in [96] focuses on survivable VNE under probabilistic regional failures. It directly incorporates the stochastic nature of regional disaster events into the VN mapping. A-priori probabilistic models are used to specify physical node/link vulnerability and advanced "risk-aware" VN mappings are computed to limit the damage from large regional faults. A strategy is also proposed to improve the balance between competing resource efficiency and risk objectives.

2.6.3.3 Discussion: Survivable Virtual Network Embedding

The main objective for optimization of the presented approaches is maximizing the revenue while minimizing the total cost through minimizing the redundant resources. The approaches are mostly protection methods for link or node failures, which reserve or backup before any failure happens. Several approaches [83, 84, 85] use path protection against link failures which could provide bandwidth saving over link protection. However, path protection is more vulnerable to multiple link failures than link protection. Shared protection for the backup links or nodes is also part of a number of approaches [82, 84, 89], which saves resources compared to dedicated protection. However, it is also more vulnerable to multiple link failures.

Most works focus on single physical failure (single link or single node failures). They assume that the network failures are independent of each other and only one failure happens at a time. Approaches for multiple node failure protection are developed in [92] and in [93]. Joint failure protection is also rarely considered; so far examples are [94] and [95]. Multiple node or link failures occur at the same time in the network and the correlations between node/link failures are not yet addressed in many approaches. The design of efficient multi-failure VN recovery schemes is a key concern as only a few initial solutions have been proposed here. Furthermore, most approaches focus on solving the survivable embedding problem in a single physical provider environment. Survivability in a multi-domain VN environment could have new challenges for inter and intra-domain link failures. Multiple simultaneous inter-domain and intra-domain failures could require developing more new mechanisms than for single domain environment.

2.7 Chapter Summary

This chapter presented the fundamentals of network virtualization, optical transport networks and data-center networks, reliability and availability and failure analysis in these networks and virtual network embedding. Further, a short survey on state-of-the-art general and survivable VNE algorithms was presented.

The failure analysis section showed which errors can occur in the different networks. In a network single and also multiple failures can occur. The single failure case happens more often than multiple simultaneous failures. Studies state that about 70% of the unplanned link failures are single link failures [39]. In data-centers link failures happen about ten times more often per day than node failures [52]. Hard disks are the most common cause of server failures. In fiber networks the number one failure reason for buried optical fiber cable is excavation, which is the damage to fiber-optic cable during an attempt to penetrate the ground.

In the following chapters the fundamentals of this chapter will be applied to develop VNE algorithms for reliability and availability and study relationship between cost saving and used embedding algorithm strategy.

3 Cost versus Virtual Link Availability in Optical Fiber Transport Networks

This chapter describes the problem of achieving high availability for service/virtual networks in optical fiber transport networks. It presents algorithms and evaluations of the cost versus availability problem in the optical fiber transport network environment.

An overview of the network model and its components is presented in Section 3.3 followed by the design of the cost model for the optical fiber transport network in Section 3.4. The method for embedding VNs with reliability and achieving a requested link availability is explained in Section 3.5. Furthermore, the method together with the developed algorithm to realize the trade-off study in the optical fiber transport network are described in detail. The developed heuristic algorithm is compared to the solution in Section 3.6. Section 3.7 presents the trade-off with a detailed parameter study. The simulation results are evaluated and further discussed to give recommendations for the (mobile) virtual network operators. The last section summarizes the evaluation results of the trade-off study.

The detailed description of the heuristic algorithm and the algorithm evaluation in this chapter has been published in [3]. The cost modeling and the trade-off study has been published in [4].

3.1 Introduction

Besides connectivity and capacity any carrier-grade virtual network has also to comply to availability targets at coping with fiber cuts and other failures. Constantly trying to minimize cost at renting (virtualized) links and nodes from Physical Infrastructure Providers (PIP) [97], the Virtual Network Operator (VNO) therefore faces a basic choice: It may build upon highly reliable network elements (nodes and links) or apply protection and restoration mechanisms on the basis of a larger number of network elements with lower availability figures and thus lower cost.

The first option – the 'high-cost physical network' approach – uses direct, shortest paths with high availability basing on high-cost links, i.e. the operator will invest in the infrastructure. The second option – the 'low-cost physical network' approach – realizes the necessary network availability in the virtual network domain by combining several parallel paths with lower availability and lower individual cost. Here, the individual physical network elements can be kept cheap while a larger number of them are required to realize the parallel paths. Thus, a

trade-off can be expected between link quality (small number of expensive paths) and capacity (combination of multiple cheap paths).

In this chapter this trade-off between the two choices of a 'high-cost physical network' approach and a 'low-cost physical network' approach focusing on the optical links is examined. Therefore, the fundamental question is tackled of whether it is better to invest in underlying infrastructure or to consider several paths (i.e. working and backup paths in parallel) to create a highly available network. This trade-off will be realized in a virtualized environment using virtual network embedding.

3.2 Network Availability Calculation and VNE with Availability in Literature

In the following the algorithm in this chapter is related to various methods and algorithms for network availability calculations and VNE algorithms with reliability known from literature.

Several authors describe the computation of the network availability in general and in the area of optical networks: [98] and [99] describe how the basic availability of a network can be calculated and provide an exact calculation [99] or analytical expressions for several different network topologies like star or crown [98]. The availability of a simple path through a network can be determined as the product of the availabilities of the components (i.e. the nodes and links) belonging to the path. Network availability is then the minimum path availability over all shortest paths between two distinct node pairs [98].
Other works focus on E2E connection availability in optical transport networks. The connection availability for different resilience mechanisms like unprotected, dedicated and shared path protection and path restoration in [29], in [50] and [79] dedicated and shared path protection and in [48] only dedicated protection is compared.
In [100], the authors examine the relation between the path availability (the product of the availabilities of the components, i.e. nodes and links that belong to the path) and the restorability of a network to dual failures (i.e. two failures present at a given time).
The problem of establishing a connection over at most k (partially) link-disjoint paths for which the availability is no less than a defined threshold is studied [101]. Instead of considering fully disjoint paths, the connection availability of partially disjoint paths is computed and an algorithm for finding (partially) disjoint paths with requested availability is provided.
These works compute or analyse the availability of an existing network with or without protection. However, they do not consider availability at the planning stage and not in virtual networks.

Lately, the research focus in VNE goes in the direction of VNE with considering the reliability of physical network explicitly.
In [102] an algorithm for VN allocation is proposed that focuses on network reliability negotiation trying to maximize the number of requests solved. They want to adapt the VN allocation according to the reliability defined by the client as well as to minimize the total bandwidth compromised to solve these requests. The algorithm generates a VN topology and calculates the network reliability of this topology. It considers the reliability of the network as the probability of the network to be operational when L failures occur. The method allows the number of

failures customization, i.e. calculates the reliability of the network considering up to L failures. The problem of reliability-aware embedding of VN where physical nodes exhibit heterogeneous and independent failure rates is studied in [103]. They solve the problem of estimating a VN's reliability by introducing the concept of "protection-domains", which represents the minimum subset of virtual nodes (primary and backups) that result in an active VN. Further, they mathematically formulate the problem of reliability-aware VNE with "just-enough reliability provisioning" in order not to waste resources through over-provisioning. The challenge is to find the trade-off between utilization efficiency of the physical network and minimizing the backups.
An approach called "reliability assurance" based on Mixed Integer Programming (MIP) problem formulation, which includes reliability requirements for VNE is developed in [104]. The VNE should fulfill strong reliability constraints while saving backup resources. The reliability of the physical network (i.e. physical routers and links) is considered during the VNE process.

Changing the underlying physical infrastructure to influence availability/reliability is not considered at network planning or VNE stage in the research work above. The authors of [105] argue that reductions in the physical Mean Time To Repair (MTTR) can also enhance availability at full or partial dual-failures - besides adding protection capacity. They show that an economic strategy exists for balancing the trade-off between capacity investment and MTTR reduction efforts to achieve high availability in networks designed to be 100% restorable against single failures. They model the cost functions for maintenance expenditures (considering the required repair time) and also the spent protection capacity and survivability mechanism. As reported in [105], with a reasonable approximation, the value of MTTR can be shown to be directly proportional to the physical unavailability of each span/link.

Now, this section considers how changes in the underlying physical infrastructure (higher MTBF values or usage of several backup paths) will interact to achieve the desired link availability of a Virtual Network Request (VNR) - the embedding at lowest cost.

3.3 Network Model

As explained in Section 2.1.2 in a network virtualization environment a VNO requests for VN on the physical network from a VNP. The VNP requests resources which meet the requirements of the VNR from the PIP who owns the physical network. The VNR from the VNO has to be mapped with the specific requirements considering survivability of the offered service. To guarantee survivability of the VNs the reliability of the physical network their components like nodes and links can be explicitly considered.

In this chapter of this thesis an explicit link availability requirement for the VNR is requested from the VNO. The idea is to embed the VNR with the requested link availability onto the physical network.

First the underlying physical network and virtual network request are described formally.

Table 3.1: Physical network input parameter

V_s	Set of physical nodes
E_s	Set of physical links
i_s	Physical node
$C_s(i_s)$	Capacity of physical node i_s
$loc(i_s)$	Location of physical node i_s
$C_s^R(i_s)$	Residual capacity of physical node i_s
$e_s(i_s, j_s)$	Physical link from node i_s to node j_s
$BW(e_s)$	Bandwidth of physical link e_s
$BW^R(e_s)$	Residual of physical link e_s
$A(e_s)$	Availability of physical link e_s
$p(i_s \rightarrow j_s)$	Physical path from node i_s to node j_s

3.3.1 Physical Network

The underlying physical network can be modeled as a graph $G_s = (V_s, E_s)$ with nodes and links. V_s and E_s represent the set of nodes and the set of links in the physical network (in this case the underlying optical transport network).

The physical nodes represent resource nodes. One resource node could be one DC or a cluster of DCs belonging to the same PIP in the same (geographical) region. $|V_s|$ denotes the number of nodes in the set of V_s. The nodes are associated with resources like CPU capacity, storage capacity etc. The resource nodes have unique geographical locations. The location of a physical node i_s is given by $loc(i_s)$. Resource node capacity is specified by C_s, e.g. the capacity of the physical node i_s is $C_s(i_s)$. The residual (actual available) capacity of a physical node $C_s^R(i_s)$ is defined as the available capacity of the physical node $i_s \in V_s$. Each physical node i_s has an availability $A(i_s)$.

The physical links represent physical connections between the physical nodes. A link between the physical nodes i_s and j_s is given by $e_s(i_s, j_s)$ and $\forall e_s(i_s, j_s) \in E_s$. Each physical link has a bandwidth resource BW and a link availability A. The link bandwidth of a link $e_s \in E_s$ is given by $BW(e_s)$. The residual (actual available) bandwidth of a physical link $BW^R(e_s)$ is defined as the total amount of bandwidth available on the physical link $e_s \in E_s$. The link availability of a physical link e_s is defined as $A(e_s)$. A path from the node i_s to the node j_s is denoted as $p(i_s \rightarrow j_s)$, which is the collection of the links along the path. The length of the path (the number of hops) is given by $|p(i_s \rightarrow j_s)|$, which is defined by the number of links along the path. P_s is the set of all feasible physical paths in G_s and $p \in P_s$. The bandwidth of a physical path $p \in P_s$ is the minimum bandwidth of all physical links e_s on the path p: $BW^R(p) = \min(BW^R(e_s)), \forall e_s \in p$.

Physical links are not reliable, therefore, the availability of links is less than 100%. In this work one assumption also is that the failures of single links are independent of each other.

In this chapter the focus is on the availability of links and complete paths because resource nodes are commonly implemented with high internal redundancy. Therefore, the assumption is that the node availability of 100%, (i.e. $A(i_s) = 1, \forall i_s \in V_s$).

3.3.2 Virtual Network Request

Table 3.2: Virtual network request input parameter

V_v	Set of virtual nodes
E_v	Set of virtual links
m_v	Virtual node
$C_v(m_v)$	Requested Capacity for virtual node m_v
$loc(m_v)$	Requested location for virtual node m_v
$e_v(m_v, n_v)$	Virtual link from node m_v to node n_v
$BW(e_v)$	Requested bandwidth for virtual link e_v
$A(e_v)$	Requested availability for virtual link e_v

A VNR from the operator can be represented as a graph $G_v = (V_v, E_v)$ which consists of virtual nodes and virtual links with node and link requirements. V_v and E_v represent the set of virtual nodes and links in the VNR. Requirements for virtual nodes are CPU capacity, storage, geographical location etc. The node capacity requirement is specified by $C_v(m_v)$ for a virtual node $m_v \in V_v$. Further, the location of a virtual node m_v is given by $loc(m_v)$. Here the important virtual link requirements are availability A and bandwidth BW. A virtual link is given by a connection between two virtual nodes m_v and n_v, $e_v(m_v, n_v) \in E_v$. The bandwidth of $e_v(m_v, n_v)$ is given by $BW(e_v)$ and the link availability by $A(e_v)$.

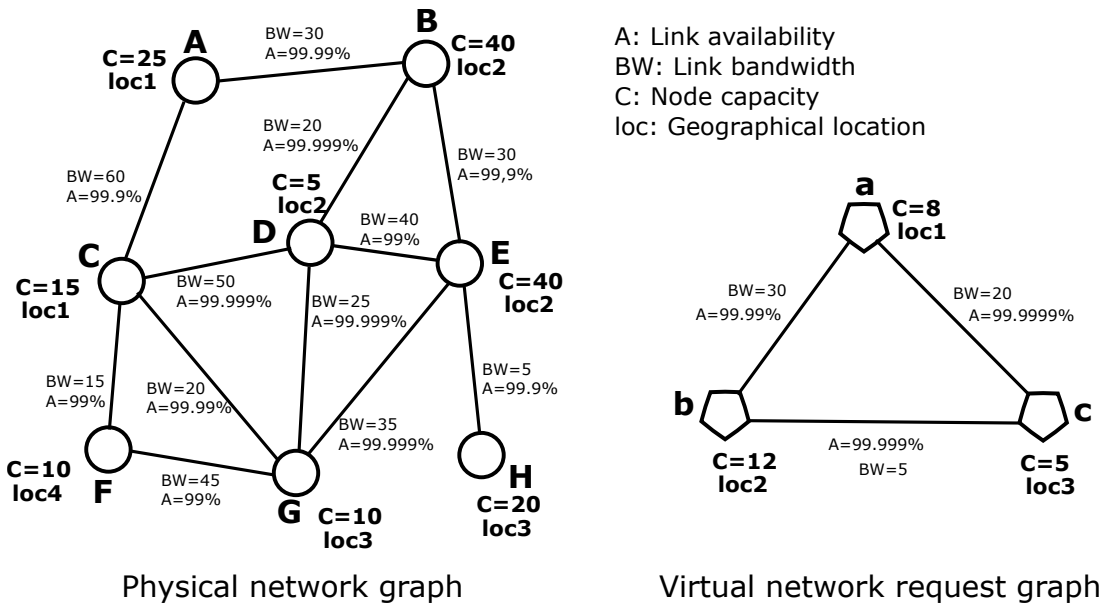

Figure 3.1: Network model with the physical network graph and the virtual network request graph

The graphical presentation of the physical network and the VNR with the important node and links constraints is illustrated in Figure 3.1. On the left the physical network graph G_s with the nodes and link resources is depicted. On the right the VNR graph with virtual link requirements is depicted. The VNR needs to be mapped on the physical network fulfilling the VNR

requirements; however, it can be seen that the virtual link availability cannot be met easily with a simple embedding of the virtual links on a path in the physical network.

3.4 Cost Model in Optical Fiber Transport Networks

One of the main requirements when building communication networks is to reduce cost. Therefore, it is important to know what cost are expected in the underlying physical optical transport network for deploying and leasing the fiber.

3.4.1 Fiber Deployment and Leasing Costs

After a detailed research about the cost for fiber deployment and fiber leasing the following results can be found.

Table 3.3: Fiber deployment cost

Fiber deployment type	Cost per kilometer
Buried	10 000 - 100 000 US$
Aerial	2 000 - 10 000 US$

For deploying buried fiber the cost range between 10 000 and 100 000 US$ per kilometer [106, 107, 108, 109]. This cost is mostly for the cable which is buried directly at 1.2m depth. The type of ground has influence on the cost of deploying buried fiber. If the cable is deployed in hard rock the cost increase and the cost decrease slightly if laid in sand ground [106].

Deploying aerial fiber results in lower cost compared to the buried fiber deployment. The cost for aerial fiber deployment range only between 2 000 and 10 000 US$ per kilometer [106, 107, 108]. However, the cost for the maintenance of the areal fiber will be much higher [106]. The cost of setting up towers in rural areas tends to be about 30-40% higher than in urban areas. The reason is that the towers need to be ground-based and consume more material [106]. In the estimation of a study it is stated that both (aerial and buried deployment) methods result in about the same cost after ten years [106].

The leasing cost for fiber vary strongly between different fiber providers as data from the Web indicate. Therefore, in this thesis the assumption for the calculations is that the cost of leasing one km of fiber for one month is 0.1% of the fiber deployment cost. Note that, in general, multiple strands of fiber are deployed during deployment. Therefore, the deployment cost can be shared among these strands. This is also reflected in current leasing fiber prices found on the Web.

3.4.2 Modeling the Relation between Fiber MTBF and Cost

This part provides a model for the relation between the fiber MTBF and cost. For the further analysis of cost versus availability problem the relation between the MTBF and availability of the underlying physical infrastructure (the fiber network) and the involved cost (i.e. the fiber deployment and leasing cost) has to be modeled in the form of a function. This function will then serve as an input for the trade-off study.

Generally, the identification of the dependency between fiber-link MTBF and the associated cost is challenging. In this case, however, the typical relationship (well-known from the field of micro-economics) between the value of an output result and the amount of an input factor applied for its increase can be observed. With increasing deployment each additional unit of the input factor will only lead to a smaller increase of the output than the deployment of the previous unit. That is, the benefit of each additional input factor unit is diminishing – more and more units of the input factor have to be spent to achieve a certain additional rise of the output. A classical example is the use of fertilizer to increase the amount of crop that can be harvested.

In the optical fiber network sheathing helps maintain fiber's usefulness over the long term. To increase the hardware fiber MTBF different sheath materials even with armor can be applied to the fiber. Depending on the sheath material and if additional armor is used cost varies with increased fiber protection measures.

Accordingly, for each additional increase of the reliability of a component (the 'output') more and more effort and cost have to be spent (the 'input'). Therefore, an exponential behavior is used to model the cost of a fiber link depending on its MTBF.

$$y = x^{\alpha} + \beta \tag{3.1}$$

Equation (3.1) represents this behavior. Here, the MTBF value is given by x and the cost is given by y. α is a scaling parameter for the curves. β is a parameter to adjust the cost curve. The relative growth of the cost according to increasing the MTBF is reflected by the scaling parameter α. The starting cost is marked by the β value. As for deploying fiber even with the lowest MTBF value, a certain amount of money needs to be spent.

As there is no known cost model up to now, the values for fiber deployment cost from Section 3.4.1 and the MTBF values for different fiber types from Table 2.2 are combined to calibrate the model. With combining the fiber deployment cost and fiber MTBF values the curves can be adjusted as in Figure 3.2 to fit the MTBF values to realistic cost per deployed fiber kilometer. In this study three cases will be considered: the first case is a linear rise in the cost, the second is a more steep rise and the last is quadratic rise (high steep with the higher MTBF values).

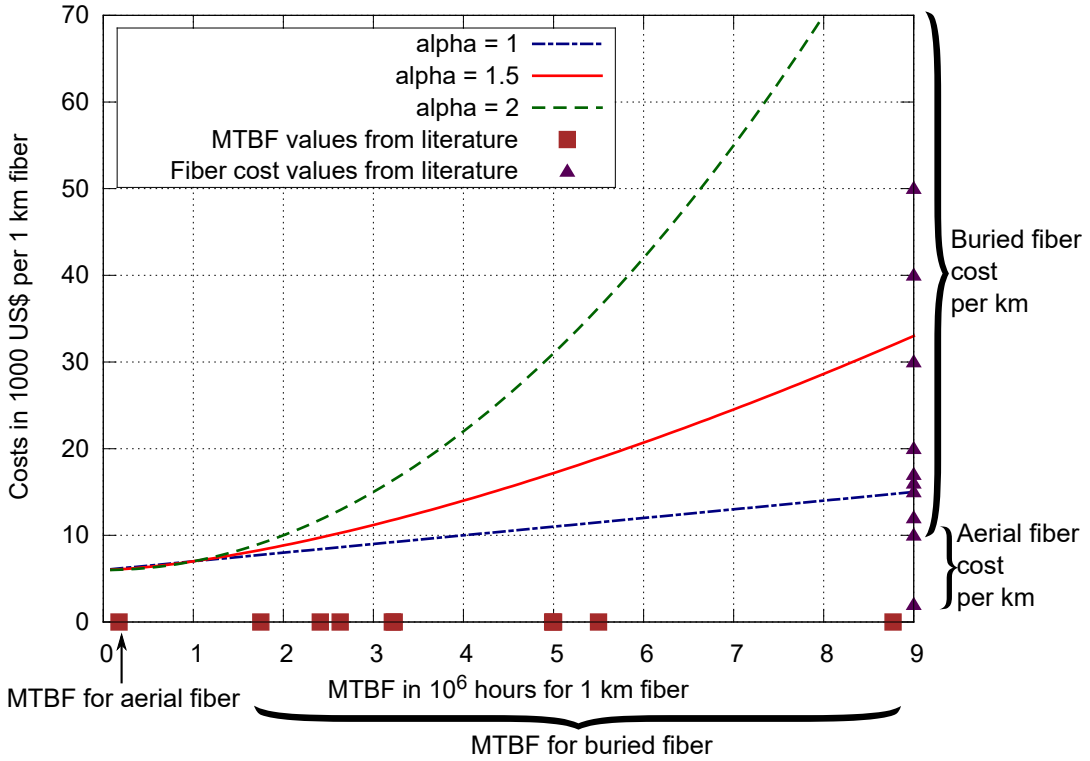

Figure 3.2: Different cost models: $y = x^{\alpha} + \beta$ [4]

3.5 Method for High Virtual Link Availability Embedding

In this section an embedding algorithm is developed for achieving high virtual link availability for the virtual network request. First, the problem is formulated mathematically. In the next step a heuristic is developed to solve the mathematical problem. Furthermore, the cost function for the cost versus availability problem in fiber networks is described.

3.5.1 VNE Problem Formulation with Link Availability Constraints

The VNE goal is to embed the VNRs in the physical network that fits the requirements of the VNR with least resources. In this case the embedding has to minimize the bandwidth resources and to satisfy the requested link availability. Therefore, a VNE algorithm is designed to provide highly available paths for the VNRs.

The VNE can be considered as a process with two stages: virtual node mapping and virtual link mapping. In the first stage virtual nodes are mapped to resource nodes in the physical network as $\mathscr{M}_{node} : V_v \mapsto V_s$. In the link mapping the feasible paths between all physical nodes mapped from the virtual nodes are established by using function $\mathscr{M}_{link} : E_v \mapsto P_s$ where $\mathscr{M}_{link}(e_v(m_v, n_v))) = p(\mathscr{M}_{node}(m_v) \rightarrow \mathscr{M}_{node}(n_v)), \forall m_v, n_v \in V_v$. Note that in this work another assumption is given: That the virtual nodes have to be in different geographical locations and are not mapped to the same physical node, i.e. each virtual node is mapped to a separate physical node.

In the mathematical formulation only a single path with the required link availability is considered for each virtual link which results in the lowest possible bandwidth usage. This VNE problem can then be formulated mathematically as a MIP as follows:

Objective:

$$\text{minimize} \sum_{e_v \in E_v} \sum_{e_s \in E_s} BW(e_v) x_{e_v e_s} \tag{3.2}$$

Link Bandwidth Constraints:

$$\sum_{e_v \in E_v} BW(e_v) x_{e_v e_s} \leq BW^R(e_s), \forall e_s \in E_s \tag{3.3}$$

$$x_{e_v e_s} \in \{0,1\}, \forall e_v \in E_v, \forall e_s \in E_s \tag{3.4}$$

Path Availability Constraints:

$$A(e_v) \leq \prod_{e_s \in E_s} A(e_s) x_{e_v e_s}, \forall e_v \in E_v \tag{3.5}$$

Path Continuity Constraints:

$$\sum_{w|e_s(u,w) \in E_s} x_{e_v e_s} - \sum_{w|e_s(w,u) \in E_s} x_{e_v e_s} = \begin{cases} 1, & u = s \\ -1, & u = t \\ 0, & u \in V_s \setminus \{s,t\} \end{cases} \quad \forall e_v \in E_v \tag{3.6}$$

Node Constraints:

$$C_v(m_v) z_{mi} \leq C_s^R(i_s), \forall m_v \in V_v, \forall i_s \in V_s \tag{3.7}$$

$$dist(loc(m_v), loc(i_s)) \leq D_{loc}, \forall m_v \in V_v, \forall i_s \in V_s \text{ and } i_s = \mathscr{M}_{node}(m_v) \tag{3.8}$$

$$\sum_{i_s \in V_s} z_{mi} = 1, \forall m_v \in V_v \tag{3.9}$$

$$\sum_{m_v \in V_v} z_{mi} \leq 1, \forall i_s \in V_s \tag{3.10}$$

The objective function (3.2) aims to minimize the overall required bandwidth for the embedding of the VNR. $x_{e_v e_s}$ is a binary variable as indicated by (3.4) denoting, whether physical link e_s is part of the mapping of virtual link e_v: 1 if true, otherwise 0. Equation (3.3) represents the bandwidth constraint that the total bandwidth of all the virtual links on the physical link e_s is limited by its bandwidth constraint $BW^R(e_s)$.

The path availability constraint is represented by Equation (3.5). This constraint is a nonlinear constraint. It calculates the path availability and ensures that the physical path, the virtual link e_v is mapped to, has equal or higher availability than the requested virtual link availability $A(e_v)$. Equation (3.6) can be viewed as path continuity constraints, where s and t are source and destination nodes of the physical path. For all the intermediate nodes on the physical path the number of incoming links is equal to the number of outgoing links.

z_{mi} is a binary variable denoting of whether virtual node m_v is mapped to physical node i_s: 1 if it is true, otherwise it is 0. Equation (3.7) represents the capacity constraint for the nodes in a

VNR. Equation (3.8) is the location constraint on the VNR. D_{loc} is a defined distance threshold. The distance between the requested location of virtual node m_v and physical node i_s, it is mapped to ($i_s = \mathscr{M}_{node}(m_v)$), should be less than or equal to the defined distance threshold D_{loc}. The distance between node m_v and node i_s is the Euclidean distance. Furthermore, Equations (3.9) and (3.10) enforce that one physical node can only accommodate one virtual node for one VNR.

3.5.2 Heuristic for Solving the Problem: Path Protection with Explicit Availability Constraints

With polynomial effort an exact solution of the previous mathematical formulation from Section 3.5.1 cannot be found. The time complexity of the previous mathematical formulation is that of an exhaustive search. For the mapping of the virtual nodes and links each possible combination would have to be calculated. However, this cannot be used in real life. Therefore, to avoid the long runtime of solving the problem optimally, this VNE problem with explicit (link) availability constraints needs to be approached heuristically. For this, a heuristic for the embedding of the VNRs is developed.

3.5.2.1 Heuristic Idea

The proposed VNE with path protection is a deterministic algorithm for finding candidate E2E paths and path pairs (i.e. primary and backup paths) that meet the availability and bandwidth requirements of the VNR.

As physical network links cannot always provide the requested availability several independent parallel links or paths of the physical network are combined to achieve the requested link availability. However, always calculating a backup path is not bandwidth efficient. For our VNE with link availability constraint not always a backup path is needed, e.g. if the primary path fulfills the availability requirements. If one single path can be found that fulfills the requested availability constraint, no backup path is required. If there is no available path within the physical network that can fulfill the requested availability constraint, multiple paths are used together to provide the requested availability. One path is used as the primary path (also called working path), the others are backups.

The mapping of virtual links to physical paths is first determined by a graph search algorithm. Here the nonlinearity of the availability constraints (as opposed to, e.g. bandwidth constraints) has to be considered: for a path consisting of x links the availability is calculated as $A_{path} = \prod_{i=1}^{x} A_i$. Therefore, if each link on the path fulfills the availability requirement A_r as $A_i > A_r$, $\forall i \in [1,x]$, the overall availability of this path is not necessarily satisfying the required link availability, i.e. $A_{path} < A_r$.

To find paths with high availability the number of components in the path should be kept low (e.g. shortest paths, fewer hops in the network). Hence, the paths using the Constrained Shortest Path First (CSPF) algorithm on bandwidth are calculated and afterwards the link availability constraint is checked. CSPF is an advanced version of shortest path algorithms. The path computed using CSPF is the shortest path fulfilling a set of constraints (in this case the requested

bandwidth per link). That means after pruning those links that violate a given set of constraints, CSPF runs shortest path algorithm. For each virtual link, k candidate paths are computed from which the most cost efficient constraint satisfying path is chosen.

Step 1:
Calculate and select a physical node candidate that fulfills the virtual node requirements (i.e. capacity and location constraint) for each virtual node.

Step 2:
Find k primary paths using CSPF algorithm on bandwidth that satisfy bandwidth requirement for each virtual link and check for these found paths if the virtual link's availability requirement is fulfilled.

Step 3:
If none of the calculated primary paths fulfills the availability constraints, find a disjoint backup path that satisfies bandwidth requirement for each found primary path.

Step 4:
Check if the primary path and disjoint path (backup path) in parallel fulfill the virtual link availability requirement.

- Yes, candidate pair found.
- No, calculate another disjoint backup path (repeat Step 3 and 4) until a combination of the primary and backup path or paths is found that fulfills the availability requirements.

Step 5:
Combine candidate primary and backup paths.

Step 6:
Use a linear program to select the best bandwidth efficient combination of path pairs out of these candidate pairs to minimize the overall bandwidth consumption for the VNE.

In the following the complete VNE algorithm is described in detail. The algorithm is split into two parts: the node mapping and link mapping.

3.5.2.2 Candidate Node Selection

The resulting node mapping is summarized in Algorithm 1. The algorithm expects the input parameters G_s and G_v, which are the graphs of the physical network and of the VNR.

For each virtual node $m_v \in V_v$, a physical node $i_s \in V_s$ needs to be found that fulfills the requirements. First the virtual nodes are sorted according to their capacity requirement C_v in descending order (line 2). The virtual node with the highest capacity constraint, node m_v, is selected first for the mapping. The reason behind this is that it is more difficult to embed a node with high capacity requirement than one with low capacity. From all physical nodes these nodes that fulfill the capacity $C_v(m_v)$ and location requirements $loc(m_v)$ of m_v are selected as possible

Algorithm 1 VNE algorithm with explicite link availability: Node mapping

procedure ALGOVNE NODE MAPPING(G_s, G_v)
  $l_{nv} \leftarrow$ sort V_v (capacity $C_v \searrow$) ▷ l_{nv} is the sorted virtual node list
  for $x \leftarrow 1, |V_v|$ **do**
    $l_{ns} \leftarrow \emptyset$ ▷ l_{ns} is the selected candidate node list
    $m_v \leftarrow l_{nv}(x)$
    for all $i_s \in V_s$ **do**
      if $C_s(i_s) \geq C_v(m_v)$ **AND** $dist(loc(i_s), loc(m_v)) \leq D_{loc}$ **then**
        $l_{ns} \leftarrow l_{ns} \cup i_s$
      end if
    end for
    if $|l_{ns}| = 0$ **then**
      return fail ▷ no possible candidate node
    end if
    $r \leftarrow 1$ ▷ r is incident link failure rate
    $b \leftarrow 0$ ▷ b is incident link bandwidth
    $n_s \leftarrow null$ ▷ n_s is 'best' candidate node
    for $y \leftarrow 1, |l_{ns}|$ **do**
      if $r(l_{ns}(y)) < r$ **OR** $\big(r(l_{ns}(y)) == r$ **AND** $b(l_{ns}(y)) > b\big)$ **then**
        $r \leftarrow r(l_{ns}(y))$
        $b \leftarrow b(l_{ns}(y))$
        $n_s \leftarrow l_{ns}(y)$
      end if
    end for
    if $n_s \neq null$ **then**
      $\mathscr{M}_{node}(m_v) \leftarrow n_s$ ▷ 'best' node mapping found
    else
      return fail ▷ no node found
    end if
  end for
end procedure

candidates and added to the candidate list (see lines 6 to 10). The capacity of a candidate node should be greater than or equal to the capacity of the virtual node m_v. The distance between the requested location of the virtual node m_v and a candidate node i_s should be less than or equal to the defined distance threshold D_{loc}. The distance between node m_v and node i_s is the Euclidean distance.

From these candidate nodes the overall incident link failure rate is calculated. The overall incident link failure rate r of a node is the multiplication of all unavailabilities of the incident links. The unavailability value is the complement of availability value. Expressed mathematically, the unavailability of a link is 1 minus the link availability value. For example for a node n with x incident links e_1 to e_x, the overall incident link failure rate r is

$$r(n) = \prod_{i=1}^{x}(1 - A_i), \tag{3.11}$$

see Figure 3.3.

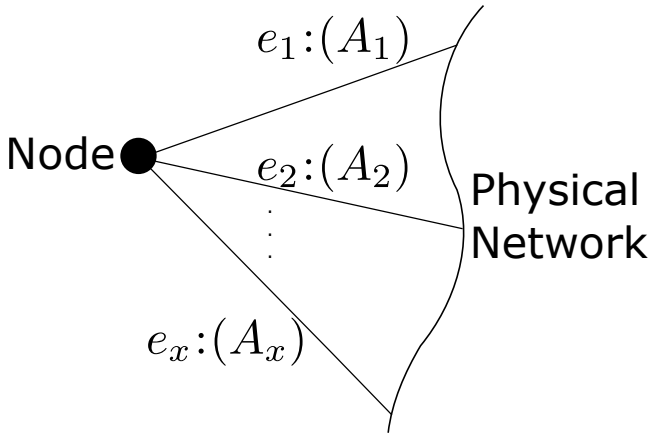

Figure 3.3: Incident link failure rate of a physical node $\prod_{i=1}^{x}(1 - A_i)$

The physical node with the lowest incident link failure rate is selected. If two or several physical nodes have the same overall incident link failure rate, the second criterion is the incident link bandwidth, so the node with the highest bandwidth of the incident links is selected. Otherwise, if no node exists that fits the virtual node requirement, a node mapping cannot be found and the algorithm stops.

After the successful mapping of all virtual to physical nodes, candidate paths and path pairs have to be found for the connection between the nodes that fulfill the requirements of the virtual links.

Figure 3.4 shows an example of the heuristic VNE algorithm for the node mapping. First the suitable physical node candidates are selected based on the capacity and location requirements for each virtual node of the VNR. After that the incident link failure rate for the candidate nodes is considered to find the 'best' node. For the mapping of virtual node **a** there are two candidates, **A** and **C**. However, **C** is selected because it has a lower incident link failure rate. Furthermore, a possible link mapping for achieving the requested link availability is depicted.

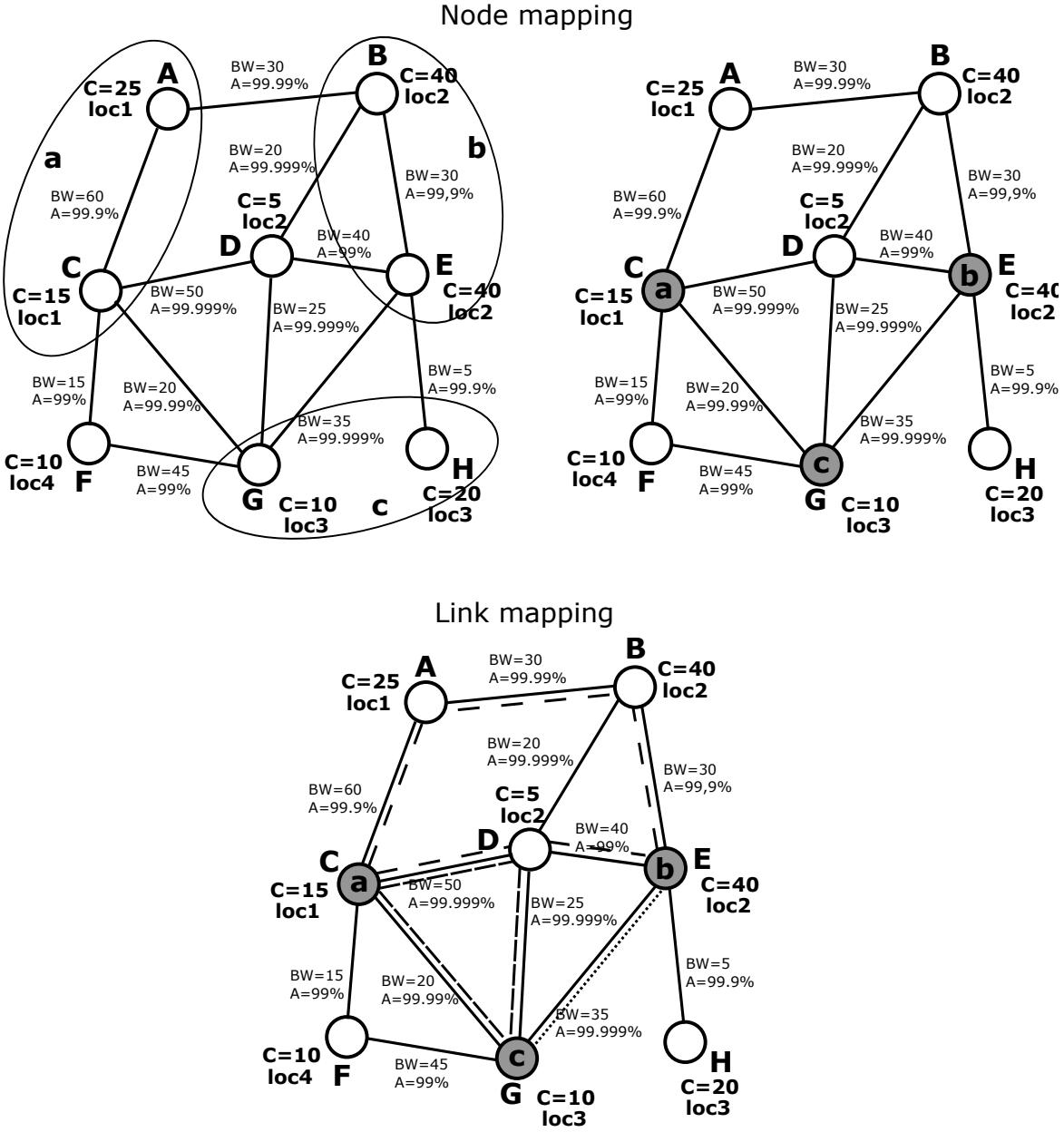

Figure 3.4: Example mapping of the virtual nodes and links for the heuristic algorithm

3.5.2.3 Link Mapping with Availability Constraint

The following describes the embedding of the virtual links in detail in two parts: the calculation of candidate paths and the path selection for the optimization.

3.5.2.3.1 Calculation of Candidates for Link Embedding

The next step is the calculation of candidate paths and path pairs for the virtual links (Algorithm 2). All virtual links are sorted according to the link availability requirement A in decreasing order. If two or more virtual links have the same availability value, they are sorted according

Algorithm 2 VNE algorithm with explicite link availability: Link mapping

procedure ALGOVNE LINK MAPPING(G_s, G_v)
  $l_{lv} \leftarrow$ sort E_v (availability $A \searrow$, bandwidth $BW \searrow$)
  for $z \leftarrow 1, |E_v|$ **do**
    $P_c \leftarrow \emptyset$ ▷ set of possible candidate paths
    $P_w \leftarrow \emptyset$ ▷ set of working paths/primary paths
    $P_{wb} \leftarrow \emptyset$ ▷ set of working-backup path pairs
    $e_v \leftarrow l_{lv}(z)$
    $P_c \leftarrow$ CALCULATECSPF(k, e_v) ▷ k- constrained shortest paths
    for all $p_w \in P_c$ **do** ▷ p_w is candidate working path/primary path
      $A(p_w) \leftarrow$ CALCAVAIL(p_w) ▷ calculate path availability
      if $A(p_w) \geq A(e_v)$ **then**
        $P_w \leftarrow P_w \cup p_w$
      else
        $p_b \leftarrow$ CALCULATEDISJOINTPATH(p_w) ▷ p_b is backup path
        if $p_b \neq \emptyset$ **then** ▷ disjoint path found
          $A(p_b) \leftarrow$ CALCAVAIL(p_b)
          $A(p_w, p_b) \leftarrow 1 - (1 - A(p_w)) \times (1 - A(p_b))$
          if $A(p_w, p_b) \geq A(e_v)$ **then**
            $P_{wb} \leftarrow P_{wb} \cup (p_w, p_b)$ ▷ add the pair (working path, backup path)
          else
            CALCULATEDISJOINTPATH(p_w, p_b)
            ▷ repeat until $A(p_w, p_{b_1} ... p_{b_n}) \geq A(e_v)$
          end if
        end if
      end if
    end for
    if ($P_w \neq \emptyset$ **AND** $P_{wb} \neq \emptyset$) **then**
      return P_w and P_{wb}
    else
      return fail ▷ no paths found
    end if
  end for
end procedure

to the bandwidth requirement BW in decreasing order (line 2). The reason behind the sorting is that links with high availability requirements are more difficult to embed compared to those with low availability requirements. For each virtual link e_v (starting with the one with the highest link availability requirement) calculate k-shortest paths as primary paths (= working paths) between the mapped nodes with the constraint on the bandwidth of the path, i.e. CSPF algorithm on bandwidth (method CALCULATECSPF with the input parameter k and the virtual link e_v). k is an operational parameter which is a number greater than or equal to 2 and can be chosen by the VNO to calculate several candidate paths. The paths found which fulfill the bandwidth requirement $BW(e_v)$ for virtual link e_v are considered as the candidates for the primary path (= working path). For each possible primary path the path availability is calculated (method CALCAVAIL, line 10).

The availability of a simple path through a network can be determined as the product of the availabilities of the components (nodes and links) that belongs to the path [98]. The E2E path availability A_{path} of a path with x links is the product of the link availability A_i of each single link e_i under the assumption that the single link failures are independent:

$$A_{path} = \prod_{i=1}^{x} A_i \tag{3.12}$$

The calculated path availability A_{path} is compared to the requested link availability $A(e_v)$. If the availability requirement is fulfilled for the primary path, this primary path is added to the set of candidate primary paths (working paths) for link e_v and no further backup path needs to be calculated.

If the path availability A_{path} is lower than the requested link availability $A(e_v)$, an additional path (backup path) is needed for fulfilling the availability requirement. A link-disjoint backup path to the existing primary path from the source to destination node fulfilling the requested bandwidth requirement is calculated using CSPF algorithm with constraints on bandwidth (method CALCULATEDISJOINTPATH, line 14).

Two paths are link-disjoint if they do not have any internal link in common. Completely disjoint paths have no intermediate element (i.e link or node) in common. Link-disjoint paths are completely disjoint paths as the unavailability of nodes is neglected here due to the fact that the nodes are commonly implemented with high internal redundancy.

In case a link-disjoint path can be found the availability constraint $A(e_v)$ must be checked for this path. Even though the link-disjoint path might not be the shortest path it could have a higher path availability than the primary path as the availability constraint is not a linear constraint. Therefore, if the availability requirement is fulfilled for the link-disjoint path (the availability of the link-disjoint path is greater than the availability of the primary path), this path becomes the primary path and is added to the set of candidate primary paths for link e_v.

In case the disjoint path has lower availability the availability of the combination of the primary (working) and backup path (former disjoint path) needs to be checked. The availability of these two paths (primary and backup path) is calculated as follows: the availability of two parallel paths is

$$A_{parallelPaths} = 1 - (1 - A_{primary}) \times (1 - A_{backup}), \tag{3.13}$$

where $A_{primary}$ is the availability value of the primary path and A_{backup} is the availability value of backup path. If the availability of primary and backup path is greater than or equal to the requested link availability $A(e_v)$, a pair (primary path, backup path) is generated and this pair is added to the set of candidate path pairs. In case the link availability $A(e_v)$ can still not be satisfied with the primary and backup path another link-disjoint backup path can be calculated to increase the availability and meet the requirement. The second backup path is calculated in the same way as the first backup path. The primary and backup paths can be combined in a way that two backup paths satisfy the virtual link availability requirement without the primary path. Then the backup path with the highest availability value is the primary path. In this way several backup paths can be calculated until the link availability $A(e_v)$ is fulfilled or no further backup path(s) can be found and no combination of the primary and backup paths satisfies the availability constraint. After successful completion of this process for all virtual links every virtual link e_v has a set of candidate paths and/or candidate path pairs.

3.5.2.3.2 Path Selection

After finding several candidate path pairs for each virtual link e_v of the VNR a bandwidth efficient combination of these candidates for the complete VNE will be calculated. Using Integer Linear Programming (ILP) the best suitable candidate path or path pair for each connection can be found and selected:

Objective:

$$\text{minimize} \sum_{e_v \in E_v} \sum_{p \in P_{e_v}} |p| BW(e_v) x_{e_v p} \tag{3.14}$$

Bandwidth Constraints:

$$\sum_{e_v \in E_v} \sum_{p \in P_{e_v}} BW(e_v) x_{e_v p} y_{e_s p} \leq BW^R(e_s), \forall e_s \in E_s \tag{3.15}$$

$$x_{e_v p} \in \{0,1\}, \forall e_v \in E_v, \forall p \in P_{e_v} \tag{3.16}$$

$$y_{e_s p} \in \{0,1\}, \forall e_s \in E_s, \forall p \in P_{e_v} \tag{3.17}$$

Link Constraints:

$$\sum_{p \in P_{e_v}} x_{e_v p} = 1, \forall e_v \in E_v \tag{3.18}$$

$$\sum_{e_v \in E_v} x_{e_v p} \leq 1, \forall p \in P_{e_v} \tag{3.19}$$

The objective function (3.14) minimizes the bandwidth for the VNR embedding while selecting the paths which require the least bandwidth. $P_{e_v} \in P_s$ is the set of pre-selected constrained link-disjoint candidate path pairs (primary + backup paths) for virtual link e_v. The pre-calculated constrained disjoint candidate path pairs step is done in the previous paragraph 3.5.2.3.1. The length of a path or path pair $|p|$ is the sum of the links along the path. $x_{e_v p}$ is a binary variable as indicated by (3.16) denoting whether virtual link e_v is assigned to the path/path pair p of primary and backup paths: 1 if true, otherwise 0.

Equation (3.15) ensures that the bandwidth $BW(e_s)$ of the physical link e_s is not overused by all virtual links that are mapped to a physical path/path pair. y_{e_sp} is an indicator (3.17) which is 1 if e_s is part of the path/path pair p, and 0 otherwise.

Furthermore, Equation (3.18) and Equation (3.19) imply that only one physical path/path pair is selected out of the candidate path pairs for virtual link e_v. After a solution is found using the ILP the selected paths or path pairs are mapped to the corresponding virtual links.

3.5.3 Cost Function and Extended Algorithm for the Availability Problem in Fiber Networks

In this part the cost model is combined with the VNE algorithm for solving the problem of cost versus high availability in the optical fiber network.

The optimization objective of the cost function of the VNE is to minimize the overall link cost for the embedding. The overall link cost is defined as the sum of the physical links that are used for the embedding of all virtual links in the VNR. Each physical link has a cost which depends on the length of the physical link of the underlying physical infrastructure (the fiber network) and its MTBF value.

Objective:

$$\text{minimize} \sum_{e_v \in E_v} \sum_{e_s \in E_s} cost(e_s) x_{e_v e_s} \tag{3.20}$$

Bandwidth Constraints:

$$\sum_{e_v \in E_v} BW(e_v) x_{e_v e_s} \leq BW^R(e_s), \forall e_s \in E_s \tag{3.21}$$

$$x_{e_v e_s} \in \{0,1\}, \forall e_v \in E_v, \forall e_s \in E_s \tag{3.22}$$

Path Availability Constraints:

$$A(e_v) \leq \prod_{e_s \in E_s} A(e_s) x_{e_v e_s} \prod_{i_s \in V_p} A(i_s), \forall e_v \in E_v \tag{3.23}$$

The objective function (3.20) aims to minimize the overall link cost for the embedding of the VNR. The physical link cost $cost(e_s)$ is defined as:

$$cost(e_s) := cost_{fiberMTBF}/1000 \times distance(e_s) \tag{3.24}$$

The $cost_{fiberMTBF}$ is the cost for deploying one kilometer of fiber with the selected MTBF value using the cost model from Section 3.4.2. For the embedding of a VNR only a fraction of the fiber is needed, i.e. virtual network embedding is like leasing fiber. Therefore, the assumption for the calculations of the cost of embedding a link on one kilometer of fiber is one thousandth of the fiber deployment cost $cost_{fiberMTBF}$.

$x_{e_v e_s}$ is a binary variable as indicated by (3.22) denoting whether physical link e_s is part of the mapping of virtual link e_v: 1 if true, otherwise 0. $distance(e_s)$ is the distance (in km) of the physical link e_s from its start node to its end node. The assumption is that the relation of distance-to-cost is linear because the VNO does not multiplex on its own (VNO would also need to rent the multiplexing equipment). Instead, only the wavelength is assigned and the distance has to be considered for each wavelength usage on the link individually.

Equation (3.21) represents the bandwidth constraint and ensures that the total bandwidth $BW(e_v)$ of all the virtual links on the physical link e_s is limited by its bandwidth constraint $BW^R(e_s)$. Equation (3.23) represents the path availability constraint. It calculates the path availability out of the availabilities $A(e_s)$ of the physical links and $A(i_s)$ of the physical nodes along the path and ensures that the path for the virtual link e_v has equal or higher availability than the requested link availability $A(e_v)$. V_p is the set of physical nodes that are along the physical path p including intermediate, start and end node of the path ($V_p \in V_s$).

After defining the cost function the embedding algorithm can be used to solve the object function which uses the cost models as input to map the VNR to the physical network.

For solving the problem of getting the lowest embedding cost while examining the trade-off between high-cost direct paths (high MTBF) and a primary path with backup path(s) to achieve the desired link availability, the algorithm from Section 3.5.2 can easily be extended. The algorithm is modified with the following extensions. First, for the physical network the link availability is calculated for all links and the MTBF-to-cost model is applied to the links (link cost) which are needed as input for the algorithm. The link availability of the physical fiber link is calculated with the Equation (2.3) and Equation (2.4) for calculation of the fiber MTBF using the distance (in km) between source and destination node. For the E2E path availability calculation in the fiber network the intermediate nodes on the path (here the switching element OXC) from the start to the end point have to be considered. For these OXC nodes in the fiber network the availability can be determined using the OXC values from Section 2.5.2.3. For a path consisting of x physical links and z physical nodes the availability is calculated as

$$A_{path} = \prod_{j=1}^{x} A(e_s)_j \prod_{k=1}^{z} A(i_s)_k \tag{3.25}$$

where $A(e_s)_j$ is the link availability of the physical link $(e_s)_j$ and $A(i_s)_k$ is the node availability of node $(i_s)_k$. Furthermore, it is considered that every physical node i_s in the network has the same node availability value (OXC availability value).

The link mapping is modified by finding a fully-disjoint backup path if the primary path availability is lower than the requested link availability of e_v. After finding candidate path pairs (primary and backup path(s)) for each virtual link e_v in the VNR, the suitable candidate for each connection using our cost function from above and the ILP is selected which results in the minimal cost for each embedding.

3.6 Path Protection with Explicit Availability Constraints: Evaluation

In this section the performance of the heuristic algorithm from Section 3.5.2 is evaluated. First, to get an evaluation of the performance of the heuristic algorithm, a number of simulations are run to compare the heuristic algorithm to the optimal MIP algorithm from Section 3.5.1 and a simple algorithm with shortest link-disjoint paths.

3.6.1 Simulation Settings

For the evaluation of the algorithm the arrival of VNRs as discrete events are simulated to compare the different algorithms. In this setting the heuristic algorithm from Section 3.5.2 uses only up to one backup path. The simple shortest disjoint paths algorithm, which will be called link-disjoint paths heuristic approach uses Suurballe's algorithm [110] to find the shortest link-disjoint paths for each virtual link. These two link-disjoint paths are used to check if the requested virtual link availability is fulfilled. All mappings of the virtual nodes to the phsyical nodes are solved using the heuristic algorithm. All linear programs are solved with glpk [111].

To evaluate the algorithm a custom-built simulation framework written in Java has been implemented. For the physical network (PN) a random graph with an average nodal degree between two and four is created. The capacity of each node is uniformly distributed within the range $[0,300]$. Further, a location denoted by x-/y-coordinates is assigned randomly to each node. Each link is assigned a bandwidth resource from the interval $[10,200]$ and a link availability between 99% and 99.999%.

For the virtual network request (VNR) the number of nodes is distributed between $[2,5]$. The VNR is a connected graph. The probability of connectivity between every two virtual network nodes is 0.5. Each node has a capacity demand between $[2,10]$ and a location constraint (i.e. x/y-coordinates). Each link requests a bandwidth demand between $[1,100]$ and a link availability between 99.9% and 99.9999%.
For both PN and VNR the resource and demand values are uniformly distributed within the defined ranges at random selections.

3.6.2 Evaluation Results

3.6.2.1 Overall Comparison

For simulations the bandwidth consumption of embedding a VNR is compared for the heuristics with the optimal results derived from exhaustive search obtained from the MIP. The MIP model is built based on the mathematical formulation at Section 3.5.1, which requires only primary path and no backup path. The non-linear availability constraint is implemented using the logarithm to eliminate non-linearity and reformulate it as a linear constraint.

Figure 3.5 shows the averaged results after 50 iterations of the bandwidth consumption of VNE obtained from different approaches using a PN with ten nodes and an average nodal degree of four. For the heuristics the same end node mapping mechanism is used to eliminate the influence

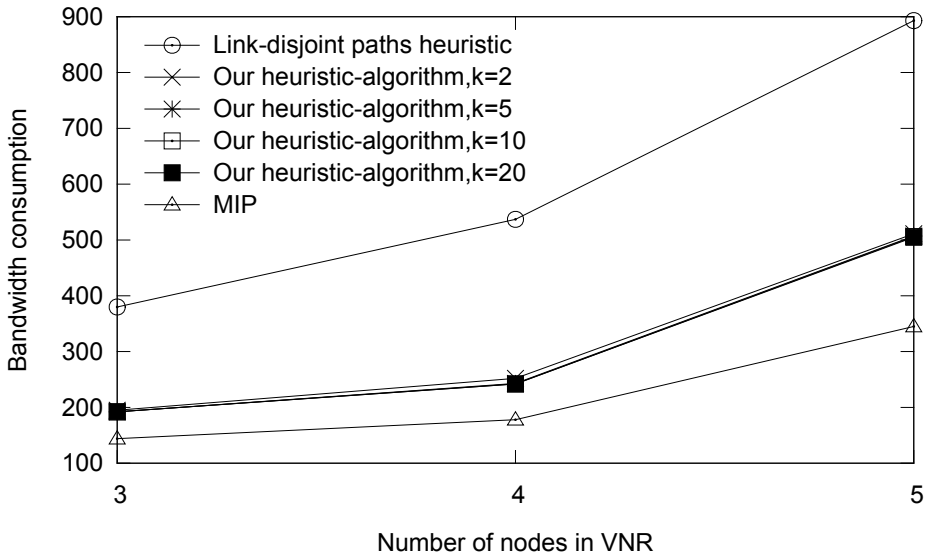

Figure 3.5: Comparison of the bandwidth consumption for the heuristics and optimal solution of the VNE with average nodal degree of 4 [3]

from the node embedding to compare the bandwidth consumption efficiency from different VNE solutions. The optimal solutions are obtained by searching the complete problem space, hence they indicate the results that are not only optimized for link mapping but also for node mapping. The optimal results are better in terms of used resources than the heuristic algorithms. However, the required computing resources and computing time cannot be neglected. For the proposal different values of k, the parameter which indicates how many primary candidates are offered to the optimization, are used. With increasing k the bandwidth consumption is getting closer to the optimal value due to selecting the best paths out of several possible candidates.

Similar results are achieved using a PN with an average nodal degree of two. In this case for the VNR the different values of k have nearly no influence on the bandwidth usage, i.e. with k=2 a near optimal solution for our heuristic can be found.

For highly available link requirements at VNR the optimal MIP algorithm which tries to only find a primary path with the requested availability value cannot be used when the PN links have low availability since short highly available paths are hardly found then. If the available link requirements at VNR are higher than the physical link availability value, it is obvious that the MIP cannot find a path since it only calculates one primary path.

3.6.2.2 Bandwidth Consumption

Bandwidth usage is one of the main metrics to evaluate the VNE algorithm efficiency. Therefore, the bandwidth consumption is compared for each VNR by using the two heuristics, the proposal (with different values for k) and the link-disjoint paths heuristic. The results in Figure 3.6 show the bandwidth consumption for embedding a VNR and indicate that our proposed algorithm consumes less bandwidth to embed a VNR compared to the link-disjoint paths heuristic. In several cases a VN link can be embedded with our heuristic using a single working path

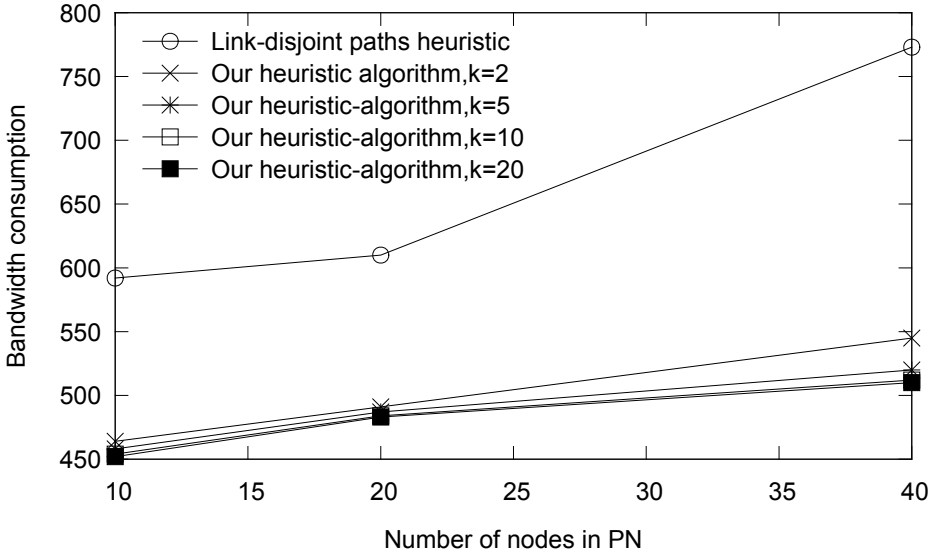

Figure 3.6: Comparison of the bandwidth consumption for the different heuristics with different requested link availabilities, number of nodes in VNR = 4, physical link availability 99% to 99.99% [3]

with high availability alone whereas in the link-disjoint paths heuristic all VN links require a backup path in the PN.

3.6.2.3 Physical Network Influence and Availability Value Influence

The topology of the PN and the requested link availability will also influence the VNE performance to a certain extent. To review this influence different sizes of nodes in the physical network are compared. The different requirements for VNR availability between 99.9% and 99.9999% were specified. The average nodal degree for the physical network is 4.

The results displayed in Figure 3.7 and Figure 3.8 are the mean values over 100 iterations in order to average the randomness effect. To obtain the VN acceptance ratio the number of accepted requests is divided by the total number of input requests. As shown in the figures our proposed algorithm has equal or higher VNR acceptance ratio for $k \geq 5$ compared to the link-disjoint paths approach for VNE. When the network size is increased, the acceptance ratios increase in both heuristics. In all cases the acceptance ratios drop if the availability constraints are tightened as the latter cannot be satisfied at all by the paths between the selected nodes. For the link-disjoint paths heuristic this is more obvious than for our proposal. The simulation results also proved that when the average nodal degree in the PN decreases, the acceptance ratios dropped in all cases. The reason for this is that it becomes more difficult to find a path or also an additional backup path to embed each virtual link.

Our heuristic has achieved good bandwidth consumption compared to an optimal solution. This heuristic can be used to provide link protection for VNE using several low-cost links (i.e. links with low availability) and achieving virtual networks for high availability requirement services.

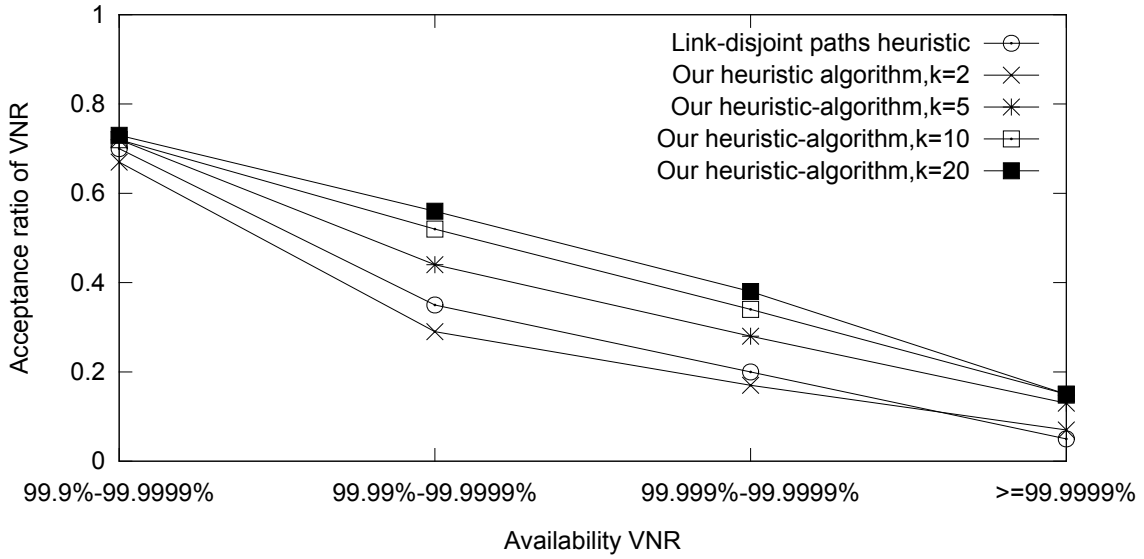

Figure 3.7: Acceptance rate of the two different heuristics, number of nodes in PN = 10 [3]

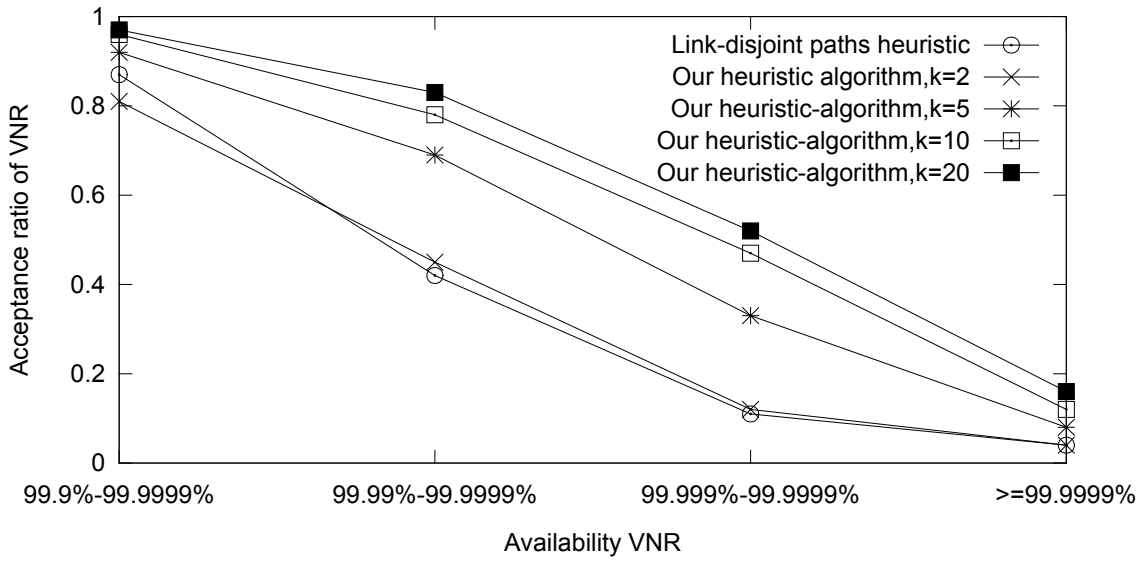

Figure 3.8: Acceptance rate of the two different heuristics, number of nodes in PN = 40 [3]

3.7 Cost versus Virtual Link Availability: Evaluation

After all the parts for the trade-off study are explained in detail the evaluation can start. This section covers the evaluation of solving the trade-off problem using the extended algorithm with the cost function described in Section 3.5.3 and the cost modeling for fiber networks from Section 3.4.2.

3.7.1 Simulation Settings

For analyzing the trade-off between a high-cost high-availability infrastructure and a lower cost redundant infrastructure the framework from Section 3.6 is extended for the scenario of optical fiber network and dealing with real-world network topologies.

The arrival of VNRs is simulated as discrete events. The input parameters for our algorithm are the MTBF values between 0.001 and 1000×10^6 hours and three different α values. As known from Section 2.5.2.3 current realistic values for MTBF are between 0.1×10^6 and 9×10^6 hours. However, the effects of very small and very large MTBF values should also be examined. The reason is to check if any abnormality exists and if using these values could result in lower physical deployment cost for the operators. The MTTR value is constant during the whole simulations and has a value of 12 hours. The α value for the cost model is chosen between 1 (linear growth) and 2 (quadratic growth). For each MTBF and α value the simulation is run 100 times and the average embedding cost are calculated.

Different physical and virtual networks are created for each simulation run. The VNR is a connected graph with 5 nodes. For the VNRs different values for the requested link availability are examined. The values are between 0.999 ('three nines') and 0.999999 ('six nines'). Further, the node requirement capacity and location and link bandwidth requirement are assigned randomly.

Table 3.4: Simulation settings

Parameters	Values
MTBF	0.001×10^6 - 1000×10^6 hours
MTTR	12 hours
α	[1, 1.5, 2]
Number of VN nodes	5
Virtual link availability	[0.999, 0.999999]

Different physical topologies are examined: a grid network structure as in Figure 3.9 and two real-world networks of different size. The area across which the nodes are distributed in the grid network is varied in size from ten kilometers to several thousand kilometers. One assumption is that each physical node and link has sufficient capacity since the focus is on the availability in this study. It should be guaranteed that at least one path can be found for the embedding when the availability requirements are met.

As the E2E physical path availability will be calculated the nodes along the path have to be considered in addition. We consider a backbone WDM network consisting of optical cross-connect nodes (OXC) interconnected by optical bidirectional links. Each link actually consists of a pair of unidirectional fibers and is capable of transporting a limited number of wavelength channels. We assume that a node is capable of routing any incoming wavelength channel on any of the incoming fibers to any wavelength channel on any of the outgoing fibers. In the case of an optical fiber network the availability of an OXC, which interconnects the optical links can also be derived from the data in [29]: With an MTBF of 1×10^5 hours and an MTTR of 6 hours

an availability of 0.99994 per OXC is reached. This OXC availability value is used in further calculations for the E2E paths.

For each virtual link k candidate paths are computed. After checking the availability constraint on the candidate paths and calculating required backup path(s) the most cost efficient combination of these candidates for the complete VNR will be calculated using Integer Linear Programming (ILP). With increasing k the bandwidth consumption/embedding cost is getting closer to the optimal value due to selecting the best paths out of several possible candidates. In Section 3.6 we proved that a value of 10 for k can achieve close to an optimum cost. Therefore, $k = 10$ is chosen in the following simulation.

3.7.2 Parameter Study: Influence of Different Parameters

In this section the influence of different parameters of the availability problem on the embedding cost is studied. One of the main questions is: What is the influence on the optimum MTBF for certain scenarios?

For these physical topologies the influence of different parameters like different cost scaling factors (i.e. α values), different extensions and different requested link availabilities by running simulations with our algorithm are examined. From the simulations the total lowest cost are determined for embedding and the corresponding MTBF value and how this cost minimum changes with the MTBF. The resulting MTBF and cost values are mean values. These values are then compared in the following sections and the results are described.

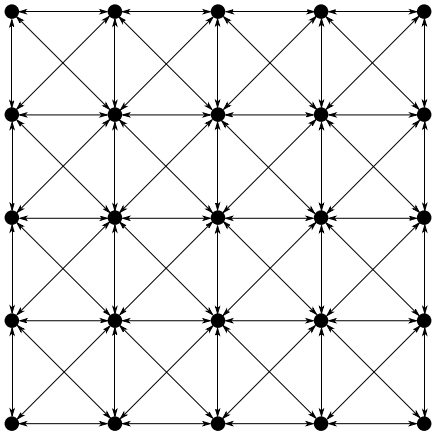

Figure 3.9: Example grid network with 25 nodes

The basis of the physical network for the simulation is a 5×5 grid topology as in Figure 3.9. A grid with a high average nodal degree is used to achieve a high acceptance ratio for the embedding and to examine the general behavior of the parameters.

3.7.2.1 Influence of the Cost Parameter α

First, the general behavior of the resulting curve is examined. The algorithm is run with the different scaling factors (α values) of the cost model and each one with the different MTBF

values. For each of the MTBF values always the lowest embedding cost value (i.e. the lowest leasing cost for deploying the VN in the fiber network) is plotted. Intuitively, we would assume a growth of cost in relation to the cost model Figure 3.2 and that the minimum embedding cost is reached at the lowest MTBF value. In Figure 3.10 an example result curve can be seen.

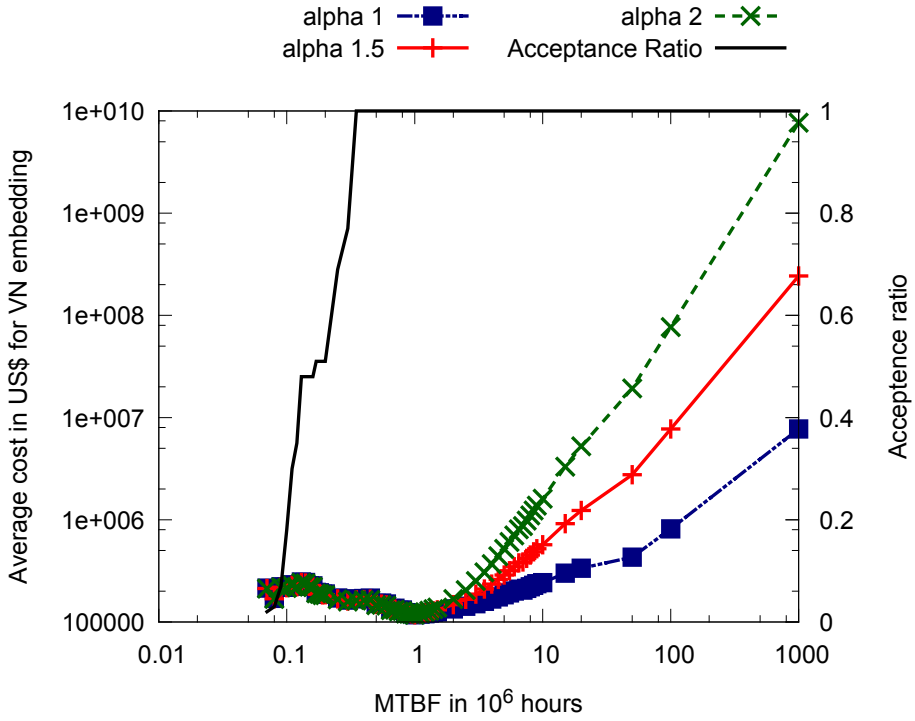

Figure 3.10: Example results of the embedding cost using different α values with a physical network of 25 nodes, area 2000km $\times$ 2000km and the VNRs with 5 nodes and requested link availability of 0.999 [4]

The cost in relation to the MTBF value is illustrated for the embedding using different α values. The acceptance ratio of the embedding is also plotted in the figure. The curve can be divided into two parts: the first part shows a decrease in the cost until the minimum. The second part is the increase of the cost. Between these two parts there is a turning point which has the lowest cost for the embedding. With increasing MTBF the curves first show a decrease of the embedding cost until a minimum is reached. The slope of the rise beyond this turning point is strongly reflecting the parameter α. For $\alpha = 1$ (linear increase of the cost) the increase is much slower than for $\alpha = 2$.

If the physical network has very low availability (MTBF values close to zero) low acceptance ratios (i.e. the percentage of successful embedding) of the embedding are the consequence. Here, the embedding cost often is very low - however, this only shows that the less complex VNRs were embedded, which themselves lead to low-cost. Thus, no real conclusion can be drawn at these low acceptance ratios. Figure 3.11 shows the zoomed in on Figure 3.10, where it can be seen that MTBF values lower than 0.07×10^6 hours cannot achieve any successful embedding. Even at 0.07×10^6 there is a local minimum of the cost which is not utilizable due to low acceptance ratio.

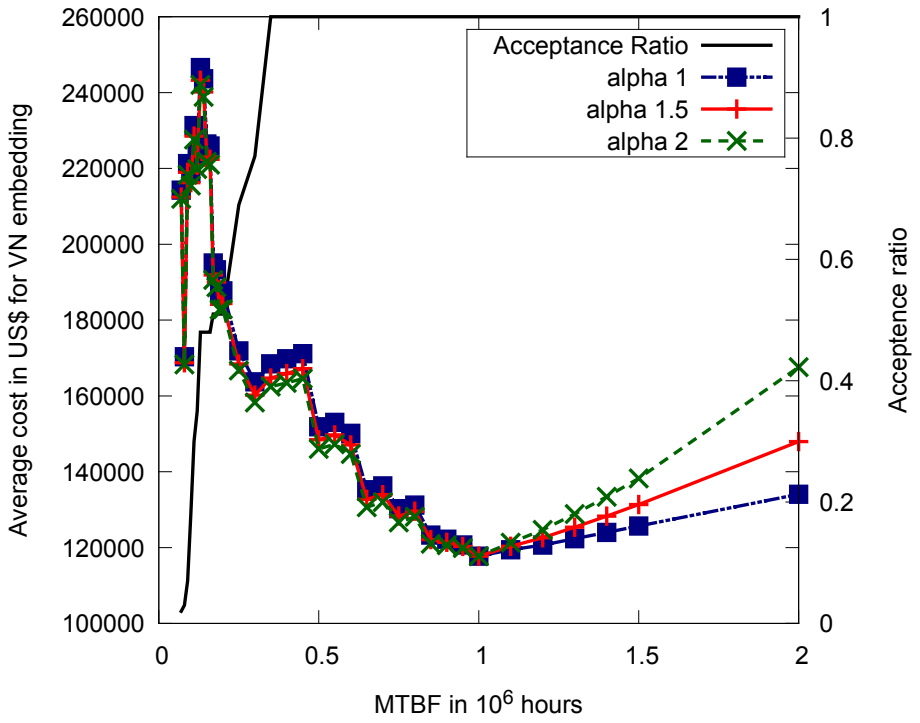

Figure 3.11: Results of the embedding cost using different α values (zoomed in on Figure 3.10) [4]

The observed turning points are generated by the number of physical paths for embedding a virtual link: For MTBF values towards zero several backup paths are needed to achieve a successful embedding for the requested virtual link availability which results in high cost. This can be seen in Figure 3.12, where up to five physical paths are needed for embedding one virtual link. For higher MTBF values fewer backup paths are required and the cost decrease. After the turning point the cost are dominated by the MTBF and rise again. In the figure, the turning point is found at a place where the MTBF values are large enough to allow the usage of only two paths (i.e. primary and backup path). It can further be noticed that while the resulting embedding cost depends on α this is not the case for the embedding decision itself (selection of the individual physical paths): For different values of α the VNRs are embedded in the same way, i.e. the same paths are selected.

Figure 3.13 shows the ratio of virtual network links that can be embedded with only a primary path. We see that until a value of about 8.5×10^6 always a backup path is needed. Compared to Figure 3.12 the same behavior can be recognized of the ratio of using primary path in relation to the MTBF values.

This section demonstrated that the minimum cost is not achieved at the lowest MTBF value but rather that there is a turning point delivering the minimum. In the next section a number of other influence factors are examined. The global minimum (turning point) is determined by simulations by changing various of the parameters. We will see that the position of the turning point depends on the topology, size and network extension of the network and the requested link availability values.

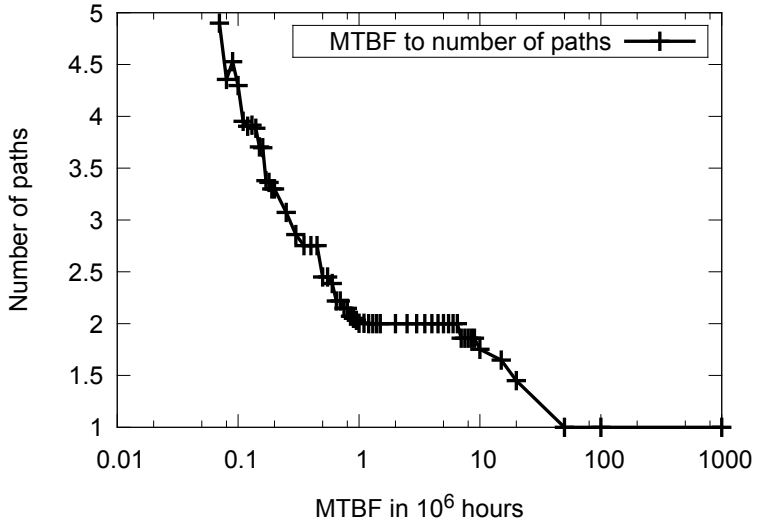

Figure 3.12: Number of (parallel disjoint) paths needed for the successful embedding of one virtual link [4]

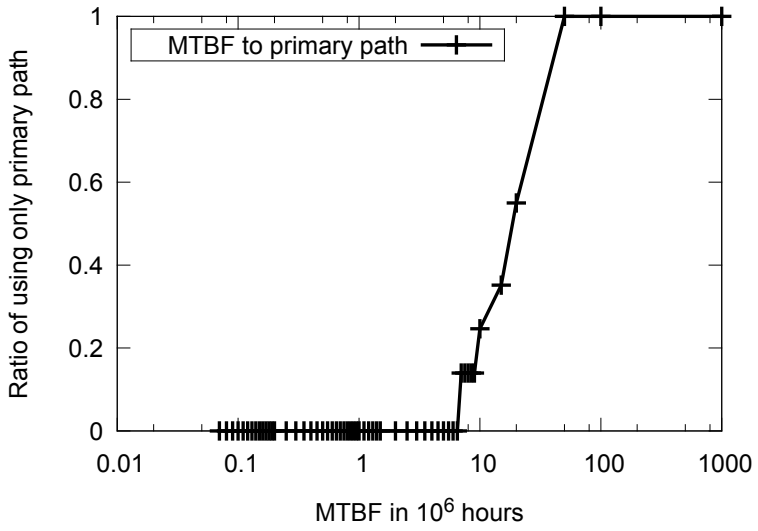

Figure 3.13: Ratio of using only the primary path for successfully embedding the links [4]

3.7.2.2 Different Network Extensions

The influence of physical network extension is investigated here. The physical network extensions are between a square area of 10×10 kilometers up to 4000×4000 kilometers. For each extension the minimum cost with its related MTBF value is extracted from the simulation results. For the example of Figure 3.10 and Figure 3.11 this is a MTBF value of 1.0×10^6.

The effect of different physical network extensions is shown in Figure 3.14 for a physical network of 25 nodes and virtual networks of 5 nodes.

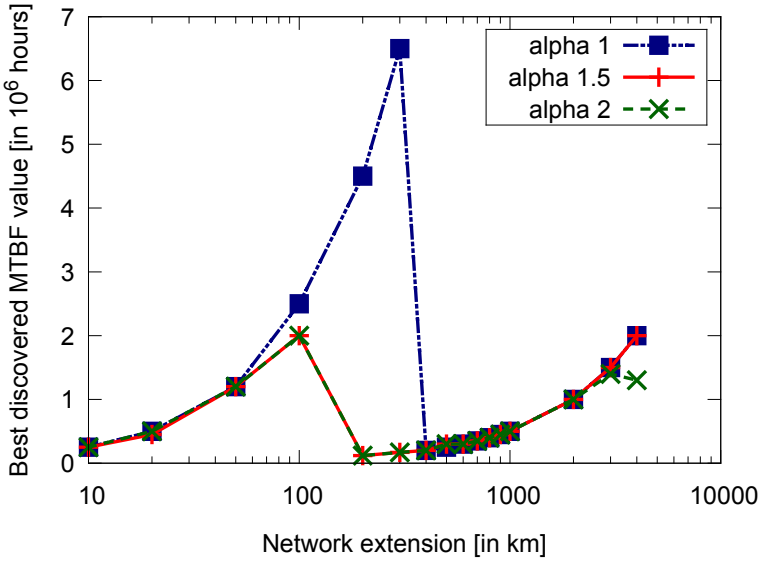

Figure 3.14: Result for different physical network extensions for 5 × 5 physical grid network (25 nodes) and VNRs with 5 nodes and a requested availability of 0.999 [4]

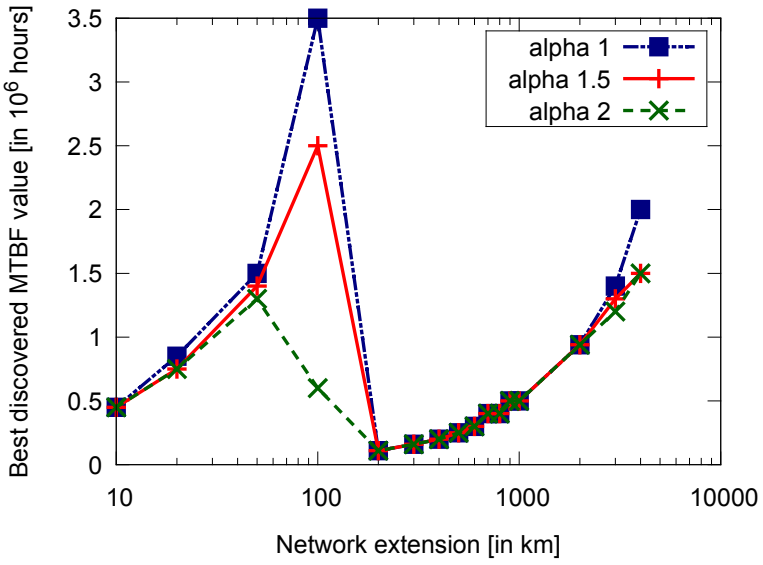

Figure 3.15: Result for different physical network extensions for 10 × 10 physical grid network (100 nodes) and VNRs with 5 nodes and a requested availability of 0.999 [4]

The result curves show the following behavior: For very small network extensions a low MTBF is sufficient to fulfill the required availability. Here, only a primary path is needed and the MTBF value with the lowest cost increases until it is not anymore economical enough. After that region there is a drop (depending on the α value at different extension values). This is the point from which onwards it is not economical to use only a primary path since the cost increase strongly for larger MTBF value. After this point it is cheaper to have backup paths than a single primary path with a high MTBF value. For the different values a number of different behavior can be seen (especially for $\alpha = 1$, linear) because in the linear cost model the cost increase much more

slowly and single paths with higher MTBF values can achieve lower cost than several backup paths. One observation is that this behavior is repeating itself for larger network extension.

Besides the dependency on the extension and on the α value a relation also to the number of nodes in the physical network can be seen in Figure 3.15, which has an influence in the region of 50 and 300 km length of the square. Compared to the results in Figure 3.14, the MTBF values with the lowest cost only go up to 3.5×10^6 and a slightly different behavior is seen for the α value.

3.7.2.3 Different Requested Link Availabilities

The effect of different requested link availabilities is now investigated. The other parameters are kept the same and again the MTBF value with the minimum cost is derived from the simulation results. The effect of higher requested link availabilities is shown in Figure 3.16. The requested link availability values are 0.999, 0.9993, 0.9996, 0.9999, 0.99999 and 0.999999.

For higher link availability values (larger than 0.999) the MTBF values with the lowest embedding cost increase. At a certain point higher MTBF values result in higher cost than an additional backup path. Therefore, a drop at highly requested availability values can be detected. This can be seen in Figure 3.16 for a value of 0.99999, where there is a drop and an additional backup path is used for the embedding. All α values show nearly the same results due to the same embedding behavior and the similar cost in the resulting MTBF range. Also a larger physical network with 100 nodes (see Figure 3.17) shows no significant change in the behavior compared to Figure 3.16 because the embedding is done similarly and identical numbers of physical paths are required for embedding the virtual links.

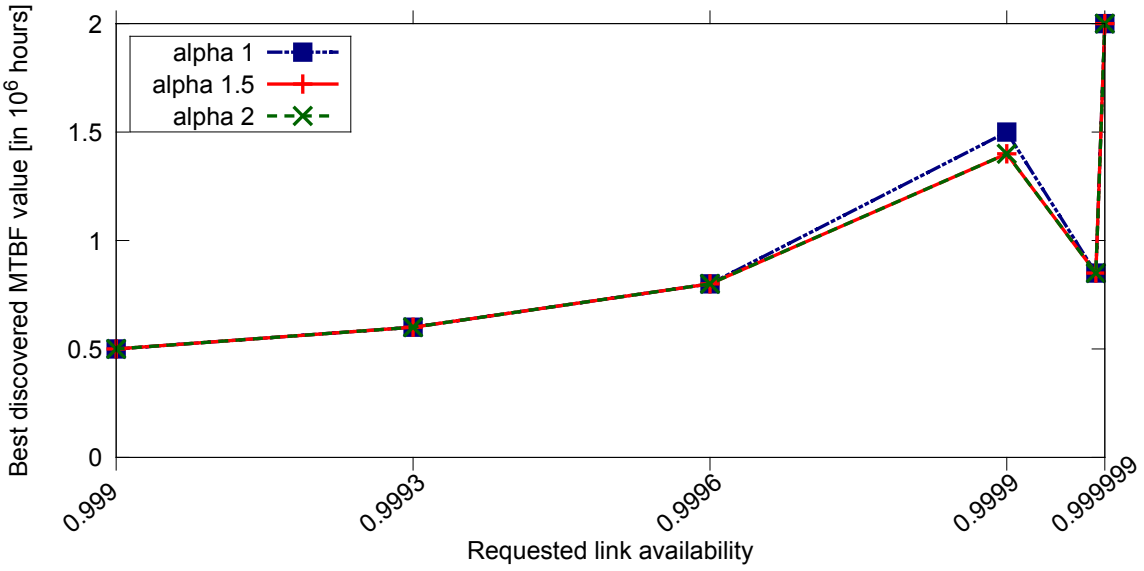

Figure 3.16: Result for different requested link availabilities using a 5×5 physical grid network (25 nodes) of the extension 1000km $\times$ 1000km [4]

In conclusion, we see that for low requested virtual link availabilities it may be cheaper to use a physical network with high MTBF values than to use backup paths. However, if higher virtual

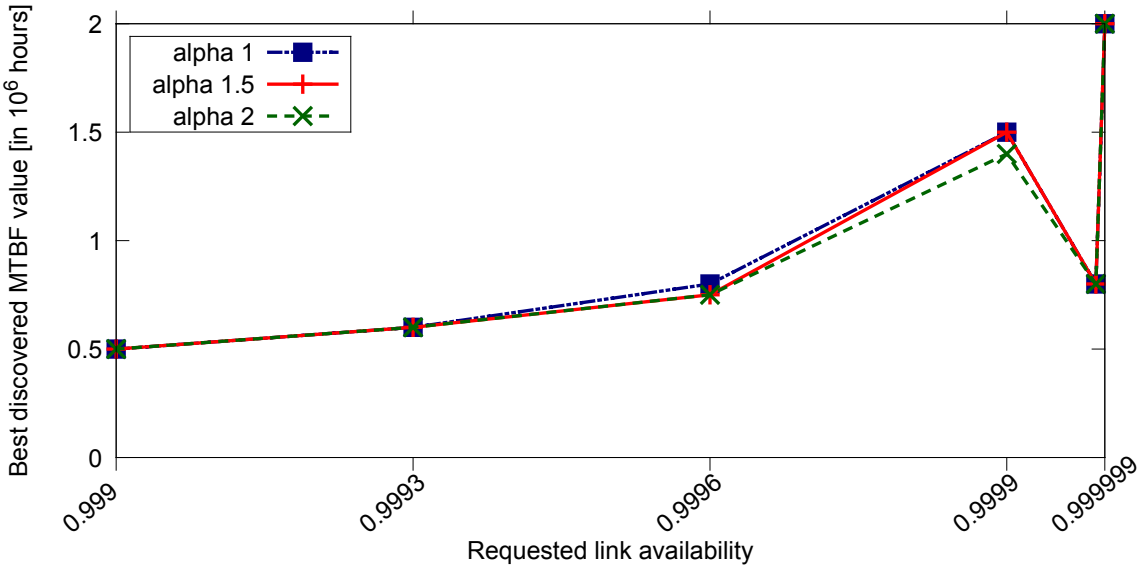

Figure 3.17: Result for different requested link availabilities using a 10 × 10 physical grid network (100 nodes) of the extension 1000km × 1000km [4]

link availability values are requested, they can only be achieved with backup paths and even higher MTBF values.

3.7.3 Real-World Network Topologies

In this section the influence of the requested link availability is examined on the basis of two different real-world networks from Germany and North America, which are publicly available network topologies from the Internet Topology Zoo [112]. In these real-world networks the cities are mapped to the nodes and their interconnections to the links of the graph.

3.7.3.1 German Network

The German network, denoted DFN, is the national IP-based research backbone network in Germany. It consists of 47 nodes and 72 fiber links, which results in an average nodal degree of 3, see Figure 3.18. The average length of a link in the German network is 116 km.

Figure 3.19 shows the behavior for requested link availabilities in the range between 0.999 (‘three nines’) and 0.999999 (‘six nines’). First, the MTBF value providing lowest cost is small and slowly increasing before rising steeply until the requested link availability reaches 0.99999. The requests can be satisfied with one primary path and a backup path with higher availabilities requiring higher MTBF values. For very high link availability values (six nines or higher), again a drop is recognized resulting from the fact that additional backup paths allow lower MTBF values. On average this results into three paths. Further, the acceptance ratio of the VN embedding is getting lower as it becomes harder to identify three paths to satisfy the required embedding constraints. Therefore, a quite low acceptance ratio of about 20% can be observed be trying to achieve the availability values of 0.999999 (‘six nines’) and identifying the lowest

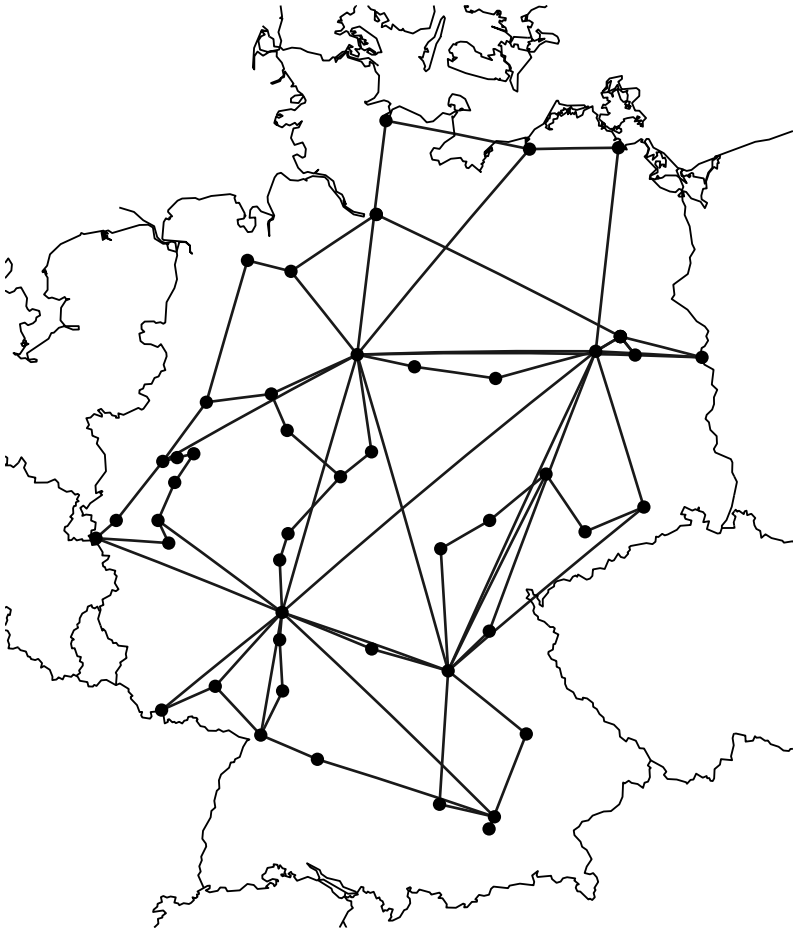

Figure 3.18: Topology of the German network [4]

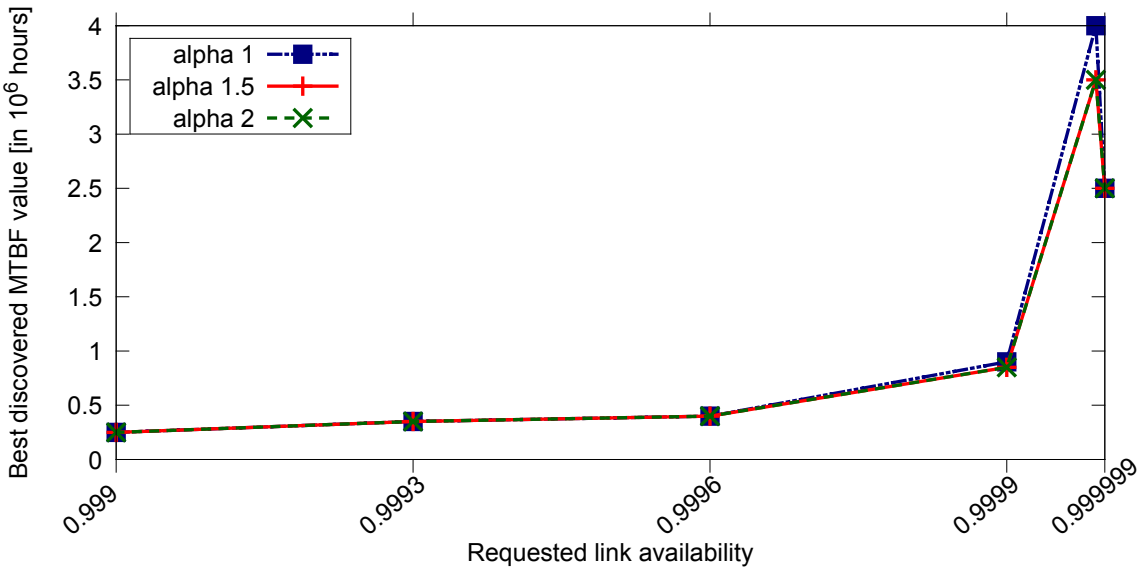

Figure 3.19: Result for different requested link availabilities using the German network as underlying physical network and VNRs with 5 nodes [4]

MTBF value. High acceptance ratios (above 60%) are only reached at MTBF values larger than 10×10^6 hours when embedding can be done with one primary and one backup path, which, however, leads to very high cost.

Using the more complex network of Germany instead of the regular grid in the previous section it can be noticed that the minimum cost can be achieved with low MTBF values (between 0.25 and 1×10^6 hours) already. However, increasing availability requirements enforces an investment in the physical infrastructure resulting in higher MTBF values.

3.7.3.2 North American Network

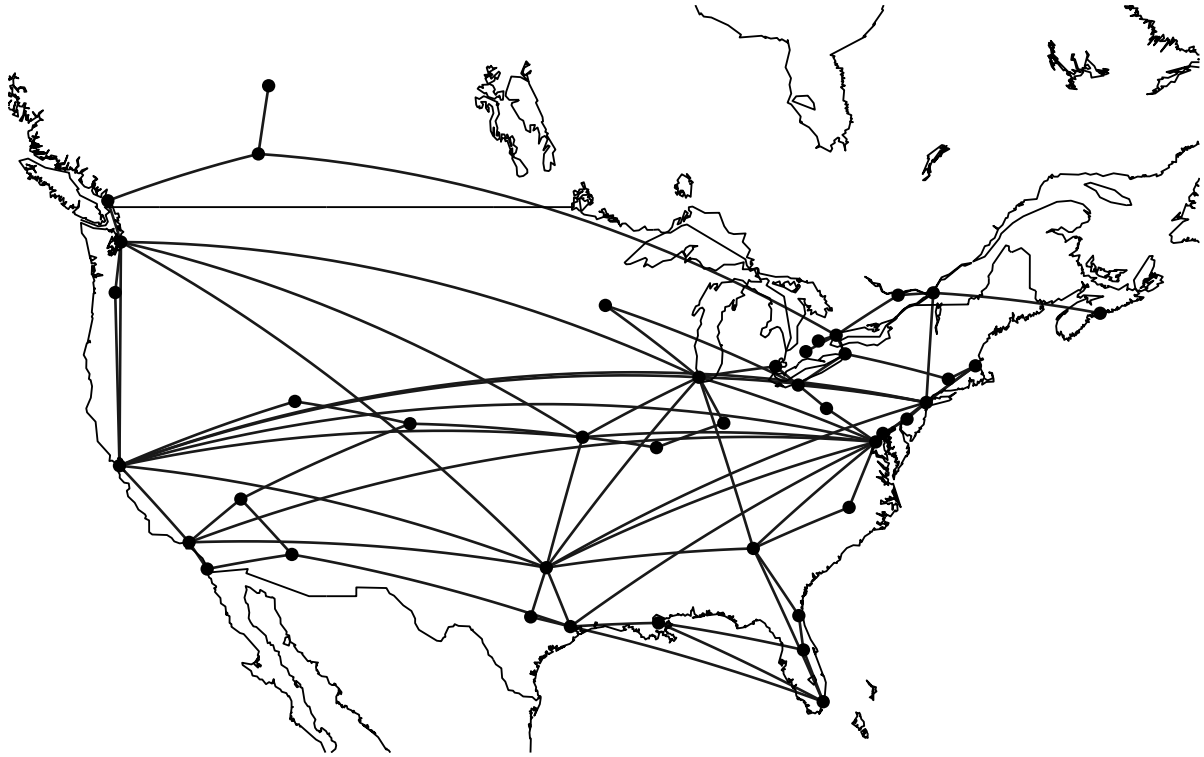

Figure 3.20: Topology of the North American network [4]

The North American network (consists of mostly USA and southern part of Canada) is an IP backbone network. It consists of 42 nodes and 77 links, which results in an average nodal degree of 3.66, see Figure 3.20. The average length of a link in the North American network is 966 km.

Figure 3.21 shows the behavior for different requested virtual link availabilities between 0.999 (‘three nines’) and 0.999999 (‘six nines’). In contrast to the previous figures an interesting different behavior (no drop in the curve) can be recognized because of the larger extension of the network and the large length of the physical links.

A strict increase in the availability to MTBF curve can be observed. First, the cost-minimal MTBF values are small and slowly increasing before they rise steeply. The requests can be satisfied with one primary path and a backup path with higher availabilities requiring higher

MTBF values. However, for availability requests of 0.99999 ('five nines') and 0.999999 ('six nines') acceptance ratio of the embedding is getting lower as it is becoming difficult to identify two paths to satisfy the required embedding constraint. Very high MTBF values are the consequence.

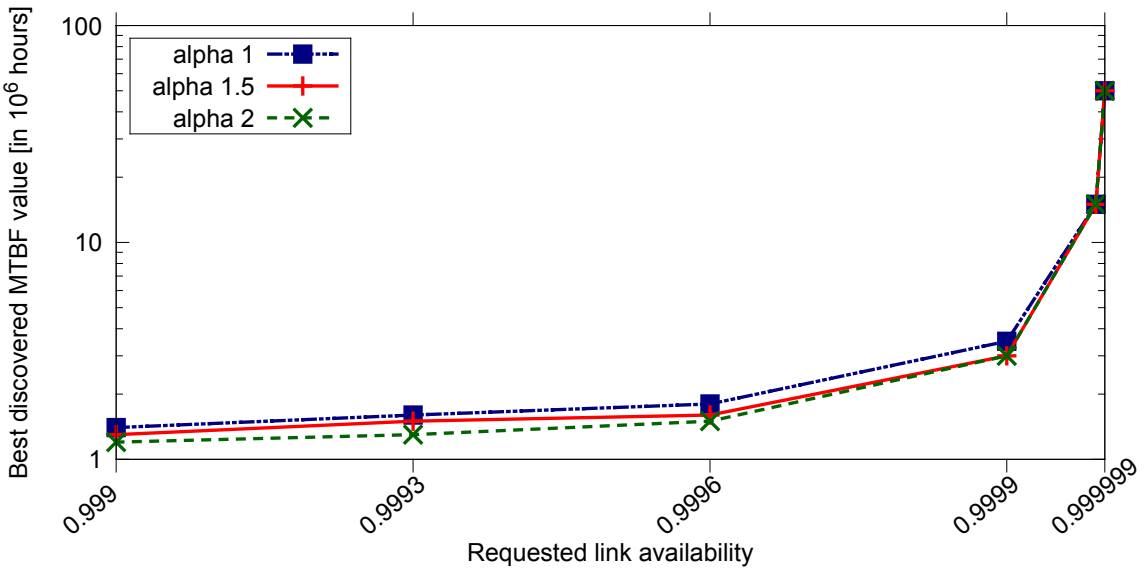

Figure 3.21: Result for different requested link availabilities using the North American network graph as underlying physical network and VNRs with 5 nodes [4]

3.7.4 Evaluation Results

This section provides a number of recommendations for operators based on our simulations.

From the results it can be seen that the cost-minimum MTBF value depends on the structure of the network as well as its geographical extension. The cost function and especially the scaling factor α play an important role. The lowest-cost MTBF value was typically in the range of 0.1×10^6 to 6×10^6 hours, which corresponds to MTBF values found in real fiber networks. MTBF values higher than this result in most cases in enormous cost.

The structure of the results shows that for an operator the turning points of the curves are of most interest - where the cost are the lowest. Whether an investment into higher MTBF values pays off mostly depends on network extension and the requested availability values. Since the increase of the availability for increasing MTBF values is logarithmic, e.g. high MTBF values like 9×10^6 hours can only achieve a four nines availability for 1 km fiber length; a lot of money has to be invested to achieve this. In contrast, combining two disjoint paths with low availability (e.g. two parallel paths with MTBF of less than 0.5×10^6 hours achieve already four nines for 1 km fiber length) already results in a high path availability. Therefore, the turning point can be observed at the least MTBF value with the least number of parallel paths. For most simulations this is in the lower range – 0.5×10^6 to 2×10^6 hours – for the MTBF values.

Depending on the requested link availabilities, the low-cost physical infrastructure approach thus can be cheaper. However, very high availability values in the region of six nines and a

moderate acceptance ratio of the embedding can only be achieved by investing in the MTBF of the physical network to get a reliable service. This study further showed that the network topology (nodal degree) and size is strongly influencing the VNE deployment strategy to achieve minimum cost. If the physical network of the operators has a low average nodal degree (e.g average nodal degree of 2), in most cases this does not allow to find disjoint paths for an embedding of any link with a requested (high) availability. Thus, in this case it is difficult or even impossible to use the low-cost physical infrastructure approach with several parallel paths here.

3.8 Chapter Summary

This chapter presented the analysis of cost versus availability in fiber transport networks. The trade-off between the two choices of a 'high-cost physical network' approach and a 'low-cost physical network' focusing on the optical links approach was examined. In the 'low-cost physical' approach as little money as possible is spent on physical protection. Instead, high availability is realized by combining multiple parallel paths to form one virtual path or link. The other design choice can be described as a 'high-cost physical' approach. Here, enough money is spent on the physical network to allow single paths to achieve a requested availability level.

In the first sections the different parts of the fiber network environment were introduced and modeled. An embedding algorithm was developed to achieve a high availability for the virtual networks on top of the physical network and as a special case on fiber transport network. The algorithm embeds virtual networks with path protection in a bandwidth efficient way while achieving the requested link availability. As the physical network links cannot always provide the requested availability, several independent parallel links or paths are combined to achieve the availability. The idea of the algorithm is to calculate the primary paths and if needed one to several backup paths which together have the requested availability.

To examine the underlying trade-off between these two philosophies a model that sets the network deployment cost in relation to the achieved resiliency was defined. Furthermore, a cost function to determine the overall cost when realizing a virtual network with a requested availability on a physical network with a different availability was created. Using the cost function in combination with the algorithm, the minimum embedding cost for an embedding can be determined. Therefore, a method was provided to possibly find an optimal strategy between Mean Time Between Failures (MTBF) increase for the physical infrastructure and additional backup paths.

Intensive simulations were done to evaluate the trade-off between cost and (link) availability when realizing virtual networks. In the simulations a number of different networks – both artificial grid topologies and real-world existing countrywide and continent-wide networks were examined. The results showed that for most configurations the 'low-cost physical' approach with low or medium reliability levels in the physical network results in the lowest cost. This is especially interesting as these reliability levels fit very well to parameters from real fiber deployments: already the lowest availability values for buried fiber found in the literature are sufficient. Therefore, it seems advisable to realize availability in the virtual domain with special protection mechanisms rather than in the physical domain in real networks.

4 Cost versus Availability for VNF Service Chains in Data-Center Networks

This chapter describes the trade-off problem between cost and element availability applied to a Data-Center (DC) network scenario. The goal is to find a topology that minimizes the cost to deploy Network Functions Virtualization (NFV) type applications and also consider the reliability especially achieving high availability for the deployed NFV services.

The first section presents an overview of the cost versus availability for Virtualized Network Function (VNF) service chains problem. In Section 4.3 the network model and its components in the DC and the VNF service chains are described in detail. The cost of different components in a DC network in connection with the cost model is explained in Section 4.4. Different strategies for the placement of VNF service chains in the DC network and methods for reliability for VNF service chains are developed in Section 4.5. These algorithms will be used to realize the trade-off study in virtualized DC environment. Finally, Section 4.6 presents the evaluation scenario and setup. Intensive evaluations of the cost versus reliability problem in the DC network environment for VNF service chains are done. The last section summarizes the chapter with the algorithm and the evaluation results of the trade-off study.

The material presented in this chapter has previously been published in [5] and [6].

4.1 Introduction and Problem Definition

NFV is a recent networking trend gaining a lot of attention from telecom operators and vendors. It promises to virtualize entire classes of network node functions within a DC and to deliver network services in the form of VNF service chains using COTS hardware and IT virtualization technologies. However, availability becomes an important issue when purpose-built telecom hardware designed for the 'five nines' standard via built-in failure protection and recovery mechanisms is replaced by the COTS hardware. With COTS DC hardware, failure probabilities could be higher than in traditional physical network infrastructure.

Today's DCs are mostly designed for services where the amount of external traffic (arriving at the DC) and the resulting traffic internal to the DC are different. Examples are outward facing

services like web type applications, or internal computing like search index calculation and data analytics where small requests can trigger large amounts of internal communication.

In comparison, NFV type applications are data intensive and require the processing of traffic streams. Here, the external traffic and the data-center internal traffic are of comparable magnitude. For NFV type application the focus is on networking and computing and not on storage.

Further, for telecom grade clouds/DCs hosting NFV type applications there are four fundamental differentiating factors that need to be considered: locality, Service-Level-Agreement (SLA) management, security and trust management and the usage of inter-cloud technologies [113]. Some telecom grade applications have very strict quality of service requirements with respect to latency and throughput; therefore, concentration becomes simply unsustainable and a more distributed, locality aware cloud infrastructure is required. Further, these applications are characterized by a number of availability and quality of service related SLAs that need to be fulfilled as the application will depend on whether and how these SLAs are met. While telecom companies are a trusted partner, cloud computing can open up for new security threats that need to be relieved like multi-tenancy, involvement of a third party (the cloud provider, in a public cloud scenario) and remote access to data and computation introduces a degree of uncertainty that needs to be mitigated.

Using the NFV concept, the perception of availability will shift from a per-network-element viewpoint to the consideration of E2E service availability. One important availability requirement in NFV is the service continuity, i.e. the E2E availability of telecommunication services. The VNF needs to ensure the availability of its part of the end-to-end service, just as in the case of a non-virtualized NF. VNF failures should never impact other applications, hardware failures should only affect those VMs assigned to that specific hardware, connectivity failures should only affect connected NFs etc. Multiple VNF components which provide the same functionality should be deployed in a parallel way into different VM to prevent single point of failure. Network operator policy for the number of redundant standby VNFs will depend on the type and criticality of the VNFs, e.g. highly critical VNFs may be set at 1+1 levels of on-site redundancy. For the on-site redundancy case redundant standby VNFs should not reside on the same servers as the operational VNFs; they should be instantiated on different servers.

As well as designing availability into a single VNF, service chains can be also designed with availability into it. Considered availability levels are in the range of the classical ‘five nines’, i.e. high availability of services, when the downtime is less than 5.26 minutes per year. However, for the purpose of the Internet of Things telecommunication networks may well have to support higher service availability values - as required, e.g. by machine control and other safety-critical applications.

The purpose is to create VNF service chains with a requested availability. As the VNF service chains are deployed in a DC an embedding algorithm for resilient deployment of VNF service chains in the DC is developed. Another important question is which DC topologies are principally best suited for the VNF service chains and especially for resilient VNF service chains. Therefore, in this chapter it will be investigated which DC topology offers the best cost-per-throughput performance for given VNF service chain availability levels.

4.2 VNF Service Chain Embedding in Data-Center Networks in Literature

Currently few research works exist on embedding of VNF service chains in DC networks. The VNF service chain embedding is a resource allocation problem similar to the VNE problem in a DC network. Even though VNE plays an important role in DC networks the embedding problem is often more focused on VM placement than on complete VNs. Furthermore, VM chain placement plays a crucial role in the layout of a VNF service chains (which are built of VMs) in a DC also.

For instance in [114] Meng et al. considered VM placement with the objective of minimizing the communication cost using traffic-aware VM placement to improve the network scalability. By optimizing the placement of VMs on servers traffic patterns among VMs can be better aligned with the communication distance between them, e.g. VMs with large mutual bandwidth usage are assigned to servers in close proximity. The placement problem has been shown to be NP-hard. The authors designed a two-tier heuristic algorithm to solve it. A comparative analysis on the impact of the traffic patterns and the network architectures (traditional DCs and recently proposed DC architectures like VL2, Fat-Tree and BCube) on the potential performance gain of traffic-aware VM placement was done. One result is that if a DC is devoted to just one application with a homogeneous traffic pattern among VMs such as a map-reduce type of workload, then traffic-aware placement of the VMs provides little improvements. The results only indicate that a BCube architecture can greatly benefit in terms of its scalability with traffic-aware VM placement while the VL2 sees the smallest benefit.
Further, in [115], two DC architectures are evaluated, FiConn (server-centric topology) and Fat-Tree (switch-centric topology), for usage of a three-tier web service application in a virtualized environment. A local VM placement scheme is compared with a service fragmentation scheme for the two DC architectures. Results showed that these two server placement schemes do not impose a significant impact on application performance in the Fat-Tree architecture. Additionally, tests with failure resilience demonstrated Fat-Tree's robustness to link/node faults.
A real-time VM allocation problem for DC, which expands the technique of Markov approximation (used in combinatorial optimization) was addressed in [116]. A joint tenant placement and route selection problem is solved by exploiting multi-path routing capabilities and dynamic VM migration. They explored how to combine VM placement and routing for DC traffic engineering and provided an efficient online algorithm for their combination.
In [117] they focus on the optimized placement of VMs to minimize the cost, the combination of network traffic cost and physical machine cost. They present an effective binary-search-based algorithm to determine how many PMs should be used, which makes a trade-off between PM-cost and network-cost.

Further resilient VM placement in DCs is studied in following papers.

For instance in [118] Xu et al. presented an optimization framework for the survivable virtual infrastructure mapping in virtualized DCs with the aim to minimize the backup resources. Like Xu et al. [118] Machida et al. [119] focus on minimizing (backup) resources, i.e. the number of redundant VMs. Thus a placement scheme for redundant VMs onto a minimal set of servers while guaranteeing a certain protection level is proposed.
The authors of [120] and [121] focus on availability-aware virtual DC embedding. The tech-

nique to compute the availability of a virtual DC in [120] considers both the heterogeneity of DC networking and computing equipment in terms of failure rates and availability and the number of redundant virtual nodes and links provisioned as backups. An allocation scheme is proposed that jointly provisions resources for virtual DCs and backups of virtual components with the goal of achieving the required VDC availability while minimizing energy cost. The authors of [122] designed an availability-aware scaling approach improving the overall system availability while maintaining the communication cost. Algorithms are used to resize up and down the VMs to meet the requirement about availability.
In [123] the aim is to answer such question as how to improve performance and availability of services hosted on Infrastructure as a Service (IaaS) clouds. Their system, structural constraint-aware virtual machine placement, supports three types of constraints: demand, communication and availability. This placement problem is formulated as an optimization problem and its hardness is proved. They design a hierarchical placement approach with approximation algorithms that efficiently solves the problem for large problem sizes. They provide a formal model for the application (to better understand structural constraints) and the DC (to effectively capture capabilities) and use the two models as inputs to the placement problem.
The work in [124] considers VNF in the cloud and presents a solution for the resilient deployment of VNFs using OpenStack for the design and implementation of the proposed service orchestrator mechanism. For service deployment resiliency the components must not be mapped to physical resources in the same fault domain. They consider component redundancy and synchronization requirements.

In contrast to the work mentioned above this chapter focuses on placement of NFV type applications with high availability constraints and the suitability of different DC for a resilient VNF service chain embedding algorithm with focus on the trade-off between cost and service availability.

4.3 System Model

In this section the different system components and their modeling will be explained in detail. The two main components are the DC and the VNF service chain. Relating to the VNE problem the DC correspond to the underlying substrate/physical network. The VNF service chain corresponds to the virtual network which needs to be embedded on the physical network, here the DC.

4.3.1 Data-Center Model

First, the DC network model which is used for the analysis will be explained in detail. Many different DC topologies have been proposed in literature. In Section 2.4 a number of the most popular (traditional and new) DC topologies are explained, which are later used for the evaluation. The intra-DC network capability depends on three factors: topology, link speeds, and switching capacity. For simplicity, only topology and link speed are considered and the assumption is that the deployed switches have sufficient switching capacity (e.g. assume all switches are full duplex with sufficient capacity in both directions). The general parameters for the DC model are as follows.

The entire DC can be modeled as a graph $G_{DC} = (V_{DC}, E_{DC})$, which consists of vertices and edges. V_{DC} and E_{DC} represent the set of vertices and the set of edges which are physical links within the DC, respectively. The nodes include switching nodes (i.e. core switches, aggregation switches and top of rack switches (ToR)/edge switches) and server nodes. An example could be the 3-tier architecture, see Figure 2.6 from Section 2.4. The physical links within the DC can be the connections between servers, servers and switching nodes and between the different switching nodes. The set of switching nodes within the DC is given by V_{Switch} and the set of server nodes (or Physical Machines (PMs)) are represented as V_{PM}, hence we have $V_{DC} = V_{Switch} \cup V_{PM}$. Each node has an availability A, with $A(j_{PM})$, $j_{PM} \in V_{PM}$ for the availability of the servers and $A(j_{Switch})$, $j_{Switch} \in V_{Switch}$ for the availability of the switches.

Each server $j_{PM} \in V_{PM}$ can be used to host one or multiple VMs. In this thesis one assumption is that all servers have the same configuration in terms of CPUs, RAM, storage and amount of cores. Thus, each server can support the same maximum number of VMs. The maximum of VMs that can be executed on each server is num_{VM}. Each server j_{PM} has a CPU capacity $C_{DC}(j_{PM})$. Each VM i on the server j_{PM} uses a percentage of capacity of $C_{DC}(j_{PM})$ for its application. Furthermore, all servers in the DC network have the same availability.

Switches are connected to other switches/servers with a specified bandwidth. Assume that each server j_{PM} is connected to a switch or another server with a bandwidth BW_p, hence all the VMs running on top of this server share this bandwidth with each other. We have $\sum_{i=1}^{num_{VM}} BW_{v_i} \leq BW_p$, where BW_{v_i} is the used bandwidth for VM i and num_{VM} is the number of VMs on a server. In theory the ToR or edge switch should be capable of supporting bandwidth of all the servers, and the aggregation switch should support the bandwidth of all the racks connected under it. In practice if this is not the case, over-subscription is considered, which refers to a point of bandwidth consolidation where the ingress bandwidth is greater than the egress bandwidth, i.e. the switch has more downlink bandwidth than uplink bandwidth.

4.3.2 VNF Service Chain Model

Second, the application that runs in the DC, the VNF service chain which is used for the analysis, will be explained in detail. A short description of what a VNF service chain is is given, followed by the general parameters for the VNF service chain model.

VNF service chains provide typical network functions like DPI (Deep Packet Inspection), firewall, encryption and tunneling to the customers of the operator offering the network service. Such service is expected to process a large number of parallel flows, where a flow is defined as all packets exchanged between two end systems that are located outside the DC. In this thesis the assumption is that the individual flows do not have inter-dependencies between each other as they, e.g. result from different customers. Therefore, the packets of one flow passing through the VNF do not influence the treatment of packets from another flow. Further, in the following the traffic needed for configuration, management and control of the VMs is neglected. Only the data traffic passing along the chain of VMs is considered.

A VNF application that runs in the DC is composed of VMs arranged in a certain logical topology that can be modeled as a network of nodes and a number of interconnecting links. The topology of such a VNF application is assumed to be a chain of 1...x VMs, i.e. one VNF after

the other. If a VNF application needs to be run in a chain of VMs, the internal topology among VMs defines the sequence that a traffic flow passes through the DC. VNFs on the VNF service chain are deployed independently on VMs, which could be located on the same server or different servers. The VNFs can be provided by the same VNF vendor or different vendors. In the VNF service chain, the flows need to traverse the function in a specific order. The task of VMs is the actual processing of the flows according to the defined function, for example performing a DPI on all packets passing through. When a traffic flow passes a VM, the VM will consume a certain amount of CPU and connectivity/bandwidth resources of the underlying server.

Depending on the running VNFs the traffic flow passing through a VM needs to be processed. For instance, the VM might drop packets from the flow or add additional header information to each packet. Thus, the output traffic load from the VM may not be the same as the input traffic load. However, for simplicity reasons, the traffic load is assumed constant in this study. In Figure 4.1 a VNF service chain is depicted with four different VNFs. The incoming packets enter the chain at the first VNF and the packets are processed in each VNF in the VMs and leave the chain. Furthermore, low latency for processing the data is needed, especially for mobile network functions.

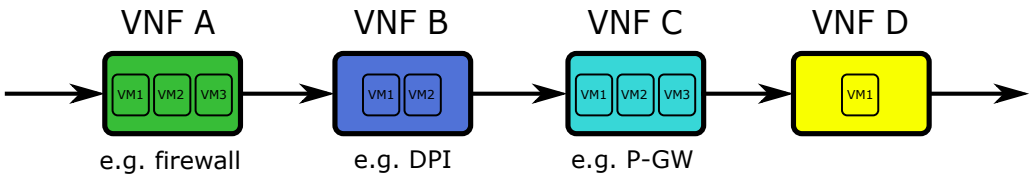

Figure 4.1: Example of a VNF service chain with 4 VNFs with different number of VMs each

Formally, a VNF service chain consisting of virtual network functions is composed of virtual machines (VMs), arranged in a certain logical topology that can be modeled as a network of nodes and a number of interconnecting links. The VNF service chain can be represented by a graph $G_{VNF} = (V_{VNF}, E_{VNF})$, where V_{VNF} and E_{VNF} represent the set of VMs and the set of links. The overall requirement for the VNF service chain/virtual network is the (service) availability A_r. When the traffic flow with traffic load f in the VNF service chain passes a VM i, it will consume a certain amount of CPU and connectivity/bandwidth resources of the underlying PM. The traffic load f can be considered as the virtual link bandwidth BW, which is considered constant in the service chain. The connection between the VMs i and j, a virtual link is given by $e_{VNF}(i,j) \in E_{VNF}$. The bandwidth/flow of $e_{VNF}(i,j)$ is given by $BW(e_{VNF})$. The CPU usage as trigged by traffic load f at VM i is denoted as $C_{VNF}(f_i)$.

4.4 Cost Model in Data-Center Networks

4.4.1 Switch and Server Cost

Important factors in order to investigate DC topologies in terms of cost are the server and switch cost. Here, only CAPEX in terms of cost is considered. However, public information for such parameters cannot be easily obtained straightforward. Therefore, several assumptions and further analysis based on available vendor data, e.g. [125, 57] are used.

The switch cost vary in price according to the corresponding speed interfaces of the switch. Today's available switches use, i.e. 10 GbE, 40 GbE or 100 GbE switch interfaces. The switches with 10 GbE interfaces can be differentiated between 10 GbE ToR switches (monolithic architecture, up to 96 ports) and 10 GbE modular switches (up to 2 048 ports).

For 10 GbE ToR switches a per-port cost of 300 US$ is chosen (out of the range 200 - 450 US$ found in the vendor data). For 10 GbE modular switches a per-port cost of 600 US$ is chosen (out of 500 - 900 US$ [125, 57]). For 40 GbE the per-port cost is usually 2-3 times higher than 10 GbE. Therefore, the selected per-port cost is 1 500 US$ at a maximum port count of 512 for this analysis. Switches with 100 GbE interfaces have nowadays a maximum port count of 192 [125, 57]. For 100 GbE a per-port cost of 6 000 US$ is chosen, which was obtained from an average of the vendor switch cost [125, 57].

In addition to the switch cost, the server cost also have to be considered. In order to highlight the bandwidth requirement for a server the server cost are modeled by two components: server blade cost and server port cost.

A server blade with ten cores can be found starting at 3 000 US$ [126]. For the different DC topologies, 10 GbE and 40 GbE server ports are required. For a 10 GbE server port the cost is 150 US$ per port. For a 40 GbE server port cost is 500 US$ per port [127].

For the DC cost modeling in this thesis, the cabling cost will not be considered as cabling cost can be regarded as very small compared to the other cost components, i.e. the switch and server cost. Table 4.1 presents a summary of the different DC components and their cost for the cost modeling.

Table 4.1: Data-center switch and server cost: The chosen values for the cost model

DC component	Cost per port	Max number of ports
Switch with 10 GbE (monolithic)	300 US$	96
Switch with 10 GbE (modular)	600 US$	2 048
Switch with 40 GbE	1 500 US$	512
Switch with 100 GbE	6 000 US$	192
10 GbE server port	150 US$	8
40 GbE server port	500 US$	4

4.4.2 The Cost Model

After analyzing the cost for the DC components, switches and servers, a cost model is needed for the analysis of the DC topologies for high availability in terms of cost.
The question regarding the switches is how the per-port prices scale. Based on the assumption that switch port cost is independent of the port count up to the possible maximum port count of the device [128], it leads to a simple linear relation between port count and switch cost. Figure 4.2 shows the result of this switch cost modeling using the values for the switches and server prices in Table 4.1.

For each DC topology the cost will be calculated using the switch cost with the help of the cost switch model in Figure 4.2 and the server cost (i.e. server blade cost plus the server port cost). As mentioned before the cabling cost will not be considered in the DC cost.

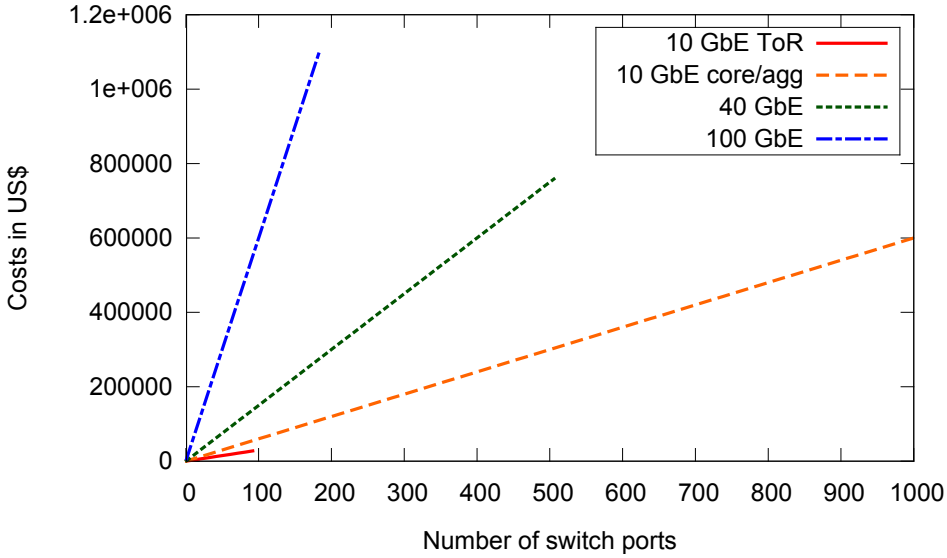

Figure 4.2: Switch cost per port [5]

4.5 Method for High Availability in VNF Service Chains

In this section the methods and algorithms for embedding the VNF service chain are explained in detail. The VNF service chains will be embedded using a VNE algorithm. In the first step the VNF service chain is embedded with a simple placement of the components of the VNF service chain (i.e. the VMs) onto the DC servers. Second, a backup strategy for the VNF service chain is developed. The last step is the complete embedding algorithm with considering the reliability and availability of the service.

4.5.1 VNF Service Chain Placement Strategies

First, the simple placement of the components of the VNF service chain (i.e. the VMs) is considered. The VNF Service Chain Placement (VSCP) strategy determines how the VNF service chain is mapped to the VM level in the DC. The applied VSCP strategy plays an important role in terms of consumed computing/storage resource and internal bandwidth of a DC. The goal is to map as many VNF service chains as possible in one DC to maximize operator's revenue. As the strategies how and where the operator will place the VNF functions in the DC are still unknown today the following three different VSCP strategies are developed:

4.5.1.1 Local VSCP

The idea of the local placement is to keep all the VMs that run VNF application sub-functions as close as possible to minimize the DC internal consumed bandwidth/number of hops for interconnecting the VMs. Since communication among VMs within the same PM does not generate network load on the physical DC network its network cost can be considered to be zero (also assuming that infinite bandwidth is available there). The idea is similar to [117], where they try to place all the required VMs on the same PM ("perfect placement"). All the servers in a DC are assigned unique identifiers (ID) according to their location in the topology, e.g. server 1 is next or closest to server 2. A list is maintained with available servers, meaning available VMs for service embedding. The available servers are sorted according to their IDs in increasing order. The server on the top of the list is selected for mapping a VNF service chain. If the resource on the selected server is not enough, the next server on the list is selected.

Advantages of the local VSCP is that all components on the chain are close to each other (e.g. even on the same server) and therefore resulting in short communication paths. Disadvantages are that if, e.g. the server or the ToR switch fails the complete chain can be down.

4.5.1.2 Random VSCP

For the random VSCP strategy a VM in the DC is chosen randomly to embed a function of a VNF service chain. The VMs can thus be on the same server or on different servers.

Disadvantages can be that all the components of the VNF service chain are distributed in the DC and the communication paths are extremely long compared to the local VSCP.

4.5.1.3 VNF Vendor-Based VSCP

A VNF service chain may contain VNFs provided by different vendors, for instance DPI from company *A* and tunneling from company *B*. In this scenario to ensure full isolation of VNFs from various vendors and to avoid the potential influence from the hypervisor and also security concerns the servers can be pooled or clustered [129].

Hypervisors can probably contain vulnerabilities that can be exploited to gain access and allow guests to break out into the hypervisor. Further, if a guest machine/VM on the same hardware as the VNF component is compromised, that compromised guest machine could be able to break into the hypervisor and then from the hypervisor compromise the VNF. To cover this use case a vendor-based VSCP strategy is introduced: A server only contains VMs from a single vendor; however, VNF components from different vendors are allowed to be on the same rack. Notwithstanding, the VMs on the same chain should be placed as close as possible to the others in the VNF service chain. Thus, the nearest server/nearest rack is selected via the lowest hop count between two servers. If two servers have the same hop count, the one with the lower ID is chosen.

Advantages are a higher degree of security from the different VNF components of various vendors since the components are not on the same server and the reliability and availability of the

VNF service chain can be higher than for the local strategy. If one server fails the other components of the VNF chains are not directly affected and using a suitable backup strategy which backs up each component separately can avoid breaking the VNF chain. Disadvantages can be that the components of the VNF chain are quite far distributed in the DC and the communication paths are longer compared to the local VSCP.

Choosing one of the VSCP strategies, a VNF service chain can be embedded as follows: The nodes of the chain are embedded according to one of the VSCP strategies (i.e. local, random or vendor-based) considering also the VNF service chain requirements of the nodes onto the suitable servers in the DC network. The links of the VNF chain are embedded using constrained shortest path routing on the requested bandwidth.

4.5.2 Backup Deployment Strategies for Reliable VNF Service Chains

The next step is to embed the complete chain in a reliable way to achieve the required service availability. Therefore, different backup strategies are designed for achieving more reliability.

The question is how to deploy VNF chains with predefined levels of availability in the DC network. As already a single failure in one part of the VNF service chain breaks the whole chain, two different backup deployment strategies are developed:

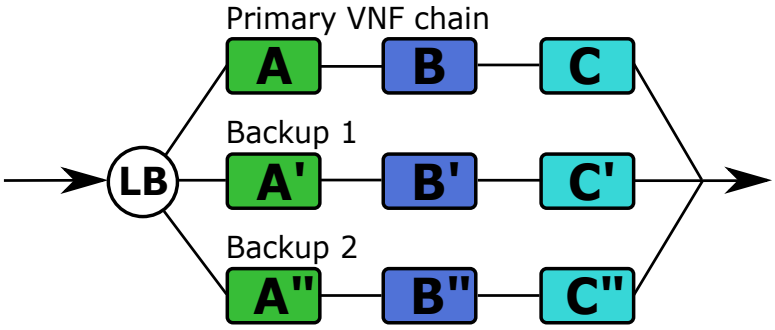

Figure 4.3: Backup deployment strategy 1 for the VNF service chain [6]

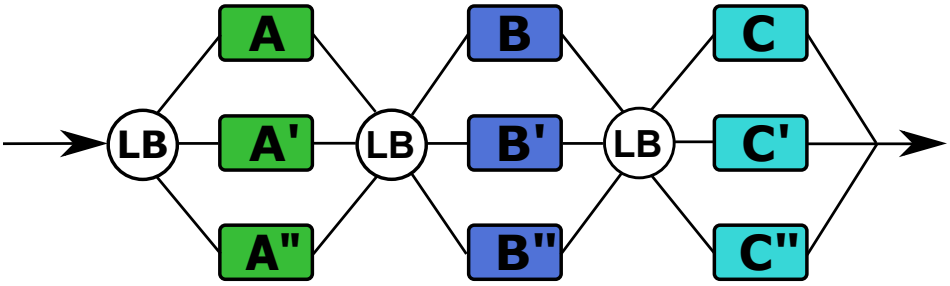

Figure 4.4: Backup deployment strategy 2 for the VNF service chain [6]

Strategy 1 is a simple backup of the complete VNF service chain via a load balancer (LB) connecting all chains at their beginning (cf. Figure 4.3 with a primary chain of three VNF functions A-B-C and two backup chains A'-B'-C' and A"-B"-C"). If a failure occurs in the VNF service chain the traffic can still be routed over the backup chains.

In strategy 2 (cf. Figure 4.4 resource pooling) the individual nodes (VNF functions) of the chains are connected to LBs that can redistribute the traffic to one of the corresponding VNFs of the backup chains if one VNF of the primary chain is broken.

The availabilities A of the different strategies are calculated as follows if the failures of the different components are independent (i.e. the different components are deployed on different physical devices):

Strategy 1

$$A = p_{LB}(1-(1-p_{VNF}^{n})^{b+1}) \tag{4.1}$$

Strategy 2 resource pooling

$$A = (p_{LB}(1-(1-p_{VNF})^{b+1}))^{n} \tag{4.2}$$

p_{LB} is the availability of the LB and p_{VNF} is the availability of a VNF component (e.g. VM). The number of VNFs in a VNF service chain is n. The number of backups per VNF component is b. The availability for strategy 1 in Equation (4.1) is calculated as the backup chains are in parallel and the components of each VNF chain are in series (multiply the availability of each VNF component). The availability for strategy 2 in Equation (4.2) is calculated as each VNF component and its backups are combined in parallel with each a LB in the front. This construct is combined in series.

4.5.3 VNF Service Chain Embedding Algorithms with High Availability

The last step is to develop a suitable algorithm for embedding the VNF service chains with reliability and high availability constraints. In this step the VSCP and the different backup deployment strategies are combined in a greedy algorithm, which calculates the required backup components for the requested VNF service availability and embeds the VNF chain. This part explains the service chain embedding algorithm for achieving high availability in detail.

4.5.3.1 Heuristic Approach

To achieve the requested service availability and save resources the smallest possible number of backup chains has to be added to the primary VNF chain. The idea is to first embed the primary VNF chain and recursively add one backup chain more while calculating the service availability. The algorithm stops if the requested availability is met or the maximum number of backup chains (e.g. 10 backups) is reached to avoid excessive numbers of intermediate switches in the backup chains. Since more backup chains consume more bandwidth direct paths for embedding the chain become harder to find the more chains have already identified and placed, resulting in long paths with many switches inbetween.

Algorithm 3 The VNF chain embedding algorithm with high availability

procedure VNFEMBEDDING(G_{DC}, G_{VNF})
  for all $i \in V_{VNF}$ in primary chain **do**
    SELECTCANDIDATENODE(i, VSCPType)
    ▷ VSCPType: local or VNF vendor-based
  end for
  for all $e_{VNF} \in E_{VNF}$ in primary chain **do**
    FINDPHYSICALPATH(e_{VNF}) ▷ find path using CSPF
  end for
  $A \leftarrow$ CALCULATESERVICEAVAILABILITY
  if $A \geq A_r$ **then** ▷ A_r is the requested service availability
    return success
  else
    $G^B_{VNF} \leftarrow \emptyset$
    while $A < A_r$ **do**
      $G^B_{VNF} \leftarrow$ CONSTRUCTBACKUPGRAPH(G^B_{VNF}, G_{VNF})
      for all $i \in V^B_{VNF}$ in backup chain **do**
        SELECTCANDIDATENODE(j, VSCPType)
      end for
      for all $e^B_{VNF} \in E^B_{VNF}$ in backup chain **do**
        FINDPHYSICALPATH(e^B_{VNF})
      end for
      $A \leftarrow$ CALCULATESERVICEAVAILABILITY
      $backup++$
      if $backup > n$ **then** ▷ n is maximum number of backup chains
        stop: **return** fail
      end if
    end while
    return success
  end if
end procedure

The resulting heuristic is summarized in Algorithm 3. The procedure VNFEMBEDDING expects the input parameters G_{DC} and G_{VNF}, which are the graph of the DC network and the graph of the VNF service chain.

The algorithm has the following steps to run through:

Step 1:
For each virtual node in the primary VNF chain calculate and select a server node candidate that fulfills the virtual node requirements (capacity, ...).

Step 2:
For each virtual link in the primary VNF chain embed it using the Constrained Shortest Path First (CSPF) algorithm with constraint on bandwidth to satisfy bandwidth requirements.

Step 3:
Construct the backup graph using one of the different backup deployment strategies.

Step 4:
Embed the backup chain while considering the constraints on CPU and bandwidth.

Step 5:
Calculate the service availability of the primary plus backup chain(s).

Step 6:
Check if the availability requirement is fulfilled

- Yes, stop the procedure and report its success.
- No, calculate another backup chain (repeat Step 4 and 5).

Step 7:
If after a certain number of backups the requested service availability is not fulfilled, stop the procedure and report its failing.

4.5.3.2 Detailed Description of the Algorithm for High-Availability Embedding

The detailed description of the complete algorithm will be given in the following.

For the identification and embedding of the backup chains first a few assumptions for simplicity are made: All servers in the DC network have the same availability. This fits the assumption that all servers have the same configuration. However, in reality this is not the case. Switches have different availabilities depending on the switch type, i.e. ToR, aggregation or core switch. The links between the switches (i.e. ToR, aggregation and core switches) are viewed as having 100% availability as only server and switch failures are considered in this thesis. Since node (i.e. switch or server) failures also affects the connection, backing up the complete VNF service chain could be seen as protecting the links indirectly and partly. LBs can be embedded on switches. Therefore, the LB has the availability of the component (e.g. switch) on which the LB is embedded. Generally, the components (i.e. the VMs) of any individual backup chain

should be placed on different servers than their counterparts in the primary and other backup chains.

At the starting point for each DC topology a so-called 'availability matrix' indicating the shared risk between any two servers is calculated. The idea is placing the VMs on different servers that do not share any common component (e.g. switch like ToR or aggregation or core switch) or least common components. Therefore, the availability of the placed VMs, especially the VNF chain, could be increased.

Each matrix element is calculated considering the probability of a failure happening in any of their common parent switches of the two servers (while entering the DC from the core switch) and the probability of a failure happening at the same time in their own private path below the common parent switches [122]. The parent nodes or switches of a server are those nodes or switches which are on the path from this server to one of the selected core switches. The availability value is calculated from the shortest distance from one server to the other. If the shortest distance is known, the number of intermediate switches (ToR, aggregation, core) can be determined. The availability A between two VMs hosted on the servers u and v is calculated as in Equation (4.3) where $C(u,v)$ is the set of the common parent switches of u and v. The sets $N(u)$ and $N(v)$ contain the parent nodes belonging to its own path (i.e. switches facing out of the DC excluding the common parent switches) of the server u and v respectively and the server itself. A_n is the availability of a node n in the DC (which could be a switch or a server).

$$A = 1 - \Big((1 - \prod_{n \in C(u,v)} A_n) + \prod_{n \in C(u,v)} A_n \times (1 - \prod_{x \in N(u), x \notin C(u,v)} A_x) \times (1 - \prod_{y \in N(v), y \notin C(u,v)} A_y)\Big) \quad (4.3)$$

For each simple VNF service chain the method SELECTCANDIDATENODE (line 4) calculates and selects server node candidates that fulfill the virtual node requirements and finally embeds the (primary) nodes of the VNF service chain on these node candidates according to the selected VSCP strategy, i.e. local or vendor-based. For the reliability case only local and vendor-based VSCP makes sense since the random VSCP would only have disadvantages of achieving high availability due to the reason that all VMs are distributed all over the DC. After the successful embedding of the nodes the method FINDPHYSICALPATH (line 7) finds and embeds the links in between the embedded nodes using an extended Dijkstra shortest path algorithm with constraints, i.e. CSPF algorithm with constraints on bandwidth.

After the simple VNF service chain has been successfully embedded the service availability A is calculated (method CALCULATESERVICEAVAILABILITY). In most cases the embedded VNF service chain does not fulfill the requested service availability A_r.
Therefore, its availability is enhanced by adding backup nodes according to one of the backup deployment strategies in Section 4.5.2. The method CONSTRUCTBACKUPGRAPH (line 15) constructs this backup graph, which needs to be embedded then. The backup nodes are embedded again according to the selected VSCP strategy for the primary VNF service chain. For each node in the backup chain suitable candidate backup servers according to the VSCP strategy and capacity constraints (i.e. CPU capacities) are identified. For each of these candidates the shared risk availability with the primary server is checked using the availability matrix explained above. The candidate node with the lowest shared risk (i.e. the highest entry in the availability matrix)

is selected and embedded. If two or more candidate nodes have the same lowest shared risk availability with the primary server, the second criterion is the shortest distance to the primary node. This continues with all backup nodes in the backup chain.

If all backup nodes are successfully embedded the backup links need to be mapped. For the local VSCP strategy the algorithm tries to embed using the shortest path that is maximum switch-node disjoint to the primary links (i.e. this means that any joint switching node contained in the primary chain should be avoided to insure less shared risked nodes and links and to increase the availability). For all other backup chains the algorithm tries to avoid joint intermediate switches between backup groups as much as possible. For the vendor-based VSCP strategy and backup deployment strategy 2 the links are mapped using shortest path and link-disjoint paths between primary backup and backup-backup links.

After embedding the first backup chain the service availability A is calculated and - if necessary (i.e. $A < A_r$) and still possible - additional backup chain(s) are determined and embedded (as described above).

This algorithm is then used in the evaluations below to compare the performance of the different DC topologies in terms of the cost per throughput relation at the required availability level of the service chain.

4.6 Evaluation

In the first part of this section the suitability and limitations of the different DC topologies for NFV type applications are examined in relation to comparing their cost. Later the high availability for the VNF application in the different DC topologies are examined. Furthermore, the cost for each different DC topology are considered again.

4.6.1 Simulation Settings

The framework for analyzing the performance of the different DC topologies is an extension to the framework in Section 3.6. The DC topologies and VNF model are implemented together with the new algorithm of VSCP and high availability VNF embedding.

The following DC parameters and VNF chain parameters are used in the simulations.

4.6.1.1 DC parameters

General DC Parameters
The DC size is determined by the amount of servers, which is within the range $[400, 4000]$ servers. Each server has 10 cores and can host up to 10 VMs. Each VM can occupy one or multiple cores within a server.

The bandwidth assignment within a DC is shown in Table 4.2.

Table 4.2: Data-center bandwidth parameters

DC bandwidth	Server-ToR	Aggregation	Core
2-tier	10 Gbps	-	100 Gbps
3-tier	10 Gbps	100 Gbps	100 Gbps
Fat-Tree	10 Gbps	20 Gbps	20 Gbps
BCube	20 Gbps	-	20 Gbps
DCell	20 Gbps	-	20 Gbps

For the 2-/3-tier tree architectures (with 24 servers per rack), four (modular) core switches are used for the topology. The other topologies (i.e. Fat-Tree, BCube and DCell) are built with low-cost switches with a low number of switch ports. Multi-stage Clos topologies built from commodity switches can support a cost-effective deployment of building-scale networks which was proved to work well in [130].
In Fat-Tree, BCube and DCell the number of core switches and the number of servers per rack are automatically determined by the structure and the number of servers.
For packet forwarding via the servers in BCube and DCell networks the packet processing capacity per core is assumed with a rate of 10 Gbps [131].

The bandwidth for each DC architecture is attempted to be distributed evenly. Fat-Tree, BCube and DCell have more links connecting the switches and servers respectively than 2-/3-tier; therefore, the required bandwidth for core and aggregation should be lower. The bandwidth values between servers and ToR switches for all the switch-centric architectures are kept the same. Lately, 10 GbE for server connectivity has been more and more adopted and will be in widespread adoption over the next few years [125, 57].

In the 2-/3-tier tree architectures the links between the servers and the ToR switches have a bandwidth of 10 Gbps and the links between ToR, aggregation and core switches have a bandwidth of 100 Gbps. For Fat-Tree the links between the servers and the ToR switches are also 10 Gbps while the links between ToR, aggregation and to core switches are 20 Gbps. For the server-centric architectures the bandwidth between the servers and switches need to be higher than for the switch-centric due to the additional packet forwarding of the server. For BCube each link has a bandwidth of 20 Gbps and $k = 1$ is set, which makes it a 2-layer switch architecture. Using two layers of switches BCube can be better compared to 2-tier architecture and the structure is less complex. For DCell each link also has a bandwidth of 20 Gbps. To achieve 20 Gbps bandwidth two 10 GbE links are used together, for instance by applying Ethernet link bundling.

DC availability parameters
The availability values of the servers and switches used in this simulation are shown in Table 4.3. These availability values are calculated using the MTBF and MTTR values from Table 2.4. For the server low-cost commodity servers are considered in all DC topologies; therefore, the lower MTBF values for the servers from Table 2.4 are chosen.

Table 4.3: Data-center component availability parameters

DC component	Availability
Server	0.999
ToR switch	0.9999
Aggregation switch	0.9999
Core switch	0.99999

4.6.1.2 VNF Chain Parameters

General VNF chain parameters
For the VNF service chain one assumption is that there are four VNFs per VNF service chain. Each of the VNFs can be built by up to 3 VMs in a row, meaning, it requires maximum 12 VMs in total to implement one VNF service chain. These 3 VMs are running different functional blocks/VNF components to implement one VNF. All the VMs are connected one after one to form a service chain, see Figure 4.1 as an example. Each VM can scale up to have more than one CPU core depending on the NFV function requirement.

The incoming packets enter the VNF chain in the first VM and traverse all the other VMs and leave the chain at the last VM. The packets are processed in each NFV function. The assumption for the traffic load is that the maximum traffic load that can be processed by a VNF service chain is 5 Gbps. If not specified differently below in the simulations, the incoming traffic load is then always 5 Gbps. If there is more than one core switch in the DC, the incoming traffic will be mostly routed into and also out of the DC using the same core switch. For each VNF chain the traffic from outside the DC reaches a core switch, is processed in the servers and has to leave at the same or in some cases (like full utilization of the core switch bandwidth) another core switch again.

Another assumption is that the required packet processing capability for each VNF is a random number between $0.65 - 2$ Gbps/core. One CPU core can forward 10 Gbps traffic in general [132]. For a typical middlebox application (e.g. a firewall) the throughput per CPU core is 2.8 Gbps for a packet size of 64 bytes and 10 Gbps for a packet size of 1024 byte [132, 133]. Other functions like carrier grade NAT, Software BRAS and Intrusion Detection System have lower throughput, only about 1 to 1.7 Gbps for a packet size of 64 byte. Packet forwarding via the servers like in BCube and DCell is assumed with a rate of 10 Gbps per core [131]. For example, if the incoming traffic load 5 Gbps and one VNF requires 1 Gbps/core processing capability, a VM with 5 cores is needed in order to process the traffic.

DC availability parameters
For the availability VNF embedding each VNF service chain is given a requested value. The requested VNF service chain availabilities can vary between 0.999, 0.9999, 0.99999 and 0.999999.

4.6.2 Comparing of the Cost for Different DC Topologies

First, the DCs with different topologies are compared according to the major cost components for DC CAPEX, i.e. the switch and server cost as shown in Figure 4.5. To compare the influence of DC topologies to the cost the number of servers in the DCs is kept the same and the total cost is calculated in order to support these amounts of servers in different DCs.

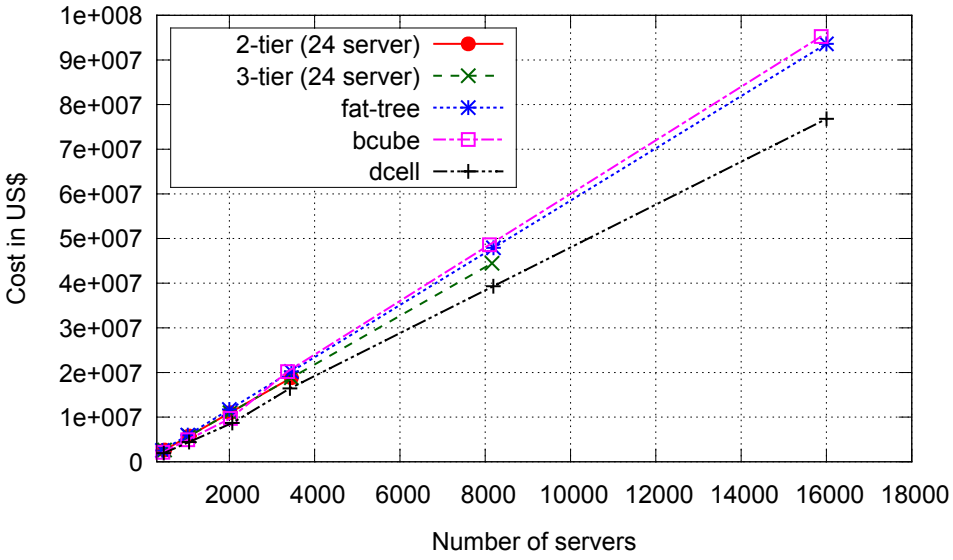

Figure 4.5: Cost of the different DC topologies with different number of servers [5]

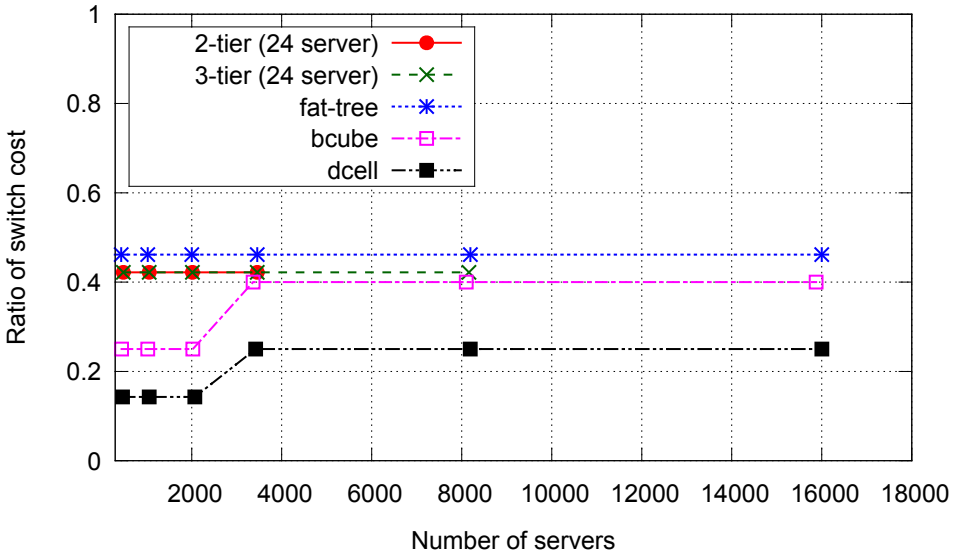

Figure 4.6: Switch cost ratio of the DC topologies with different number of servers [5]

The 2-tier architecture scales only to about 3500 servers. The 3-tier architecture scales up to about 8000 servers. Here, the 3-tier architecture has been configured with no over-subscription rate in the aggregation layer. The higher the over-subscription rate, the lower the cost will be. The DCell architecture shows the lowest cost compared to the other DC topologies, the reason of

which is further investigated in Figure 4.6, which depicts the switch cost in relation to the total cost: For BCube and DCell the switch cost are less than the server cost in relation to the total cost. In general, the server cost are more than 50% of the total cost for all the DC topologies, among which, Fat-Tree has the highest proportion and the DCell has the lowest. Therefore, the reason that the DCell architecture shows the lowest cost compared to the other DC topologies is that it requires the smallest number of switches among all architectures. Moreover, the switch cost used for DCell is much cheaper than for 2-/3-tier. The above results provide us a general understanding of DC cost and these DC architectures will be evaluated in the next section for VNF chain embedding.

4.6.3 Cost versus VNF Service Chain Embedding in DC Networks

In this part the simulation for the VNF service chain embedding versus cost is done. The results are presented comparing different parameters and their impact on the performance and cost. For each parameter the simulation is run 100 times and the average number of successfully embedded VNF chains is calculated. The VNF service chains are embedded using one of the VSCP algorithms (local, random or VNF vendor-based), i.e. the VNF service chain nodes are embedded according to the VSCP and the links of the chain are embedded using constrained shortest path algorithm on the required bandwidth.

4.6.3.1 Impact of DC Topologies

The first simulations are run with short VNF chains: 4 VNFs per chain and each VNF uses one VM, which means that one VNF service chain consists of 4 VMs. The input traffic for each VNF chain is 5 Gbps. During one simulation the VNF service chain requests are embedded one by one (by using different VSCP strategies) in a DC until the DC does not have enough resource to place further requests. Then the number of successfully embedded VNF chains for each DC topology is determined.

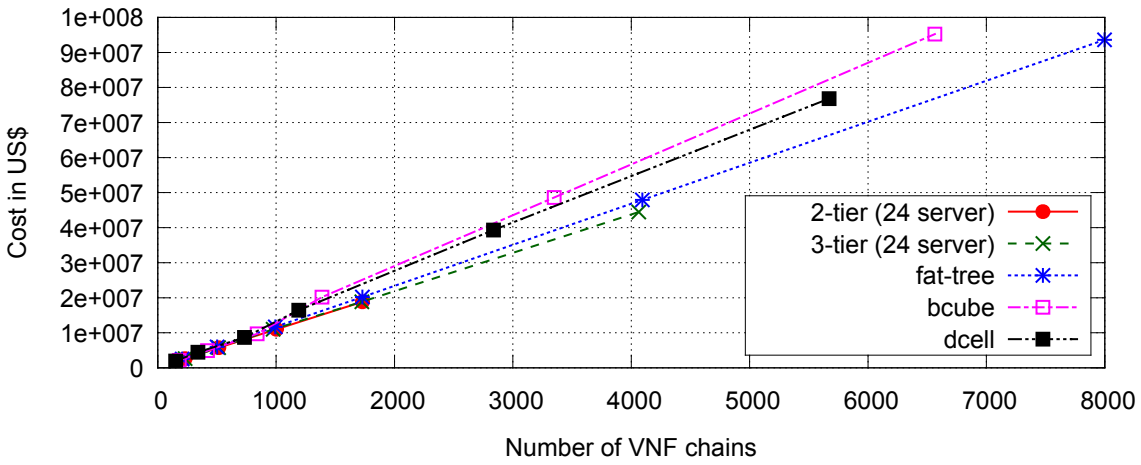

Figure 4.7: Cost vs. successfully embedded VNF chains for the local VSCP strategy [5]

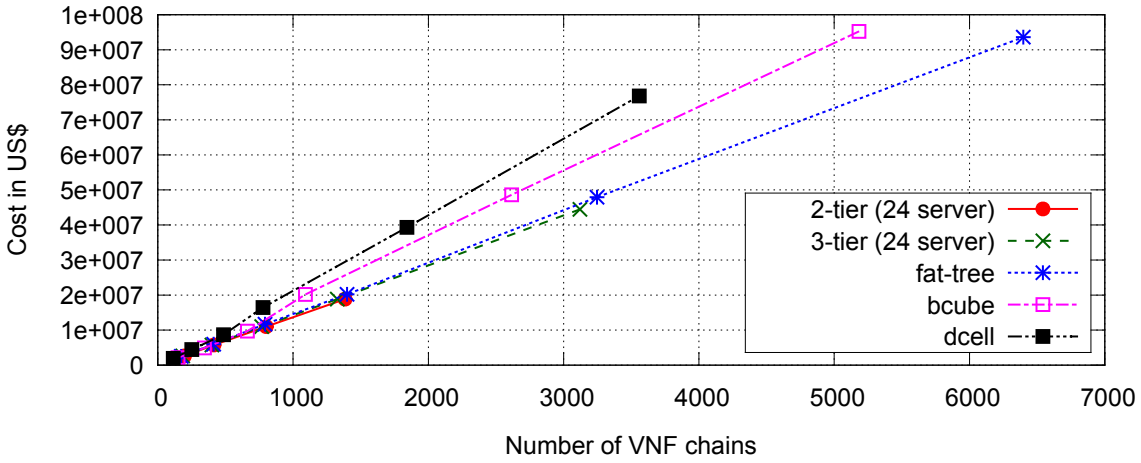

Figure 4.8: Cost vs. successfully embedded VNF chains for the random VSCP strategy [5]

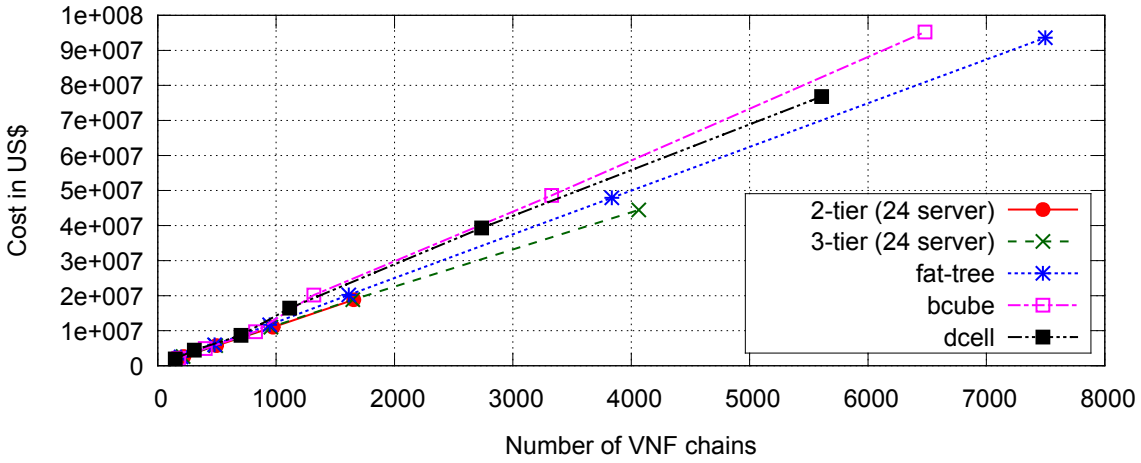

Figure 4.9: Cost vs. successfully embedded VNF chains for the vendor-based VSCP strategy

Figure 4.7, Figure 4.8 and Figure 4.9 show the results for the local, random and VNF vendor-based VSCP strategy, respectively. The x-axis indicates the number of successfully embedded VNF chains of each DC topology and the y-axis expresses the cost. For small-scale DCs the 2-/3- tier architectures have the cost advantage in terms of the number of embedded VNF chains. When comparing the embedding cost of these two architectures with Fat-Tree, BCube and DCell it can be recognized that the cost is lowest for all VSCP strategies. However, 2-/3- tier architectures are limited by the scalability issue. Further, 3-tier architecture with over-subscription in the aggregation results in a lower number of successful VNF chain embeddings. Fat-Tree‘s performance of successful embedding is close to the 2-/3- tier, especially for vendor-based VSCP. Therefore, for large-scale DCs, Fat-Tree has the highest number of successfully embedded VNF chains for the same number of servers in the DC. It has also a lower cost for the same number of embedded VNF chains compared to DCell and BCube architecture for both, local and especially random VSCP. The reason is that some percentage of the computing resources (server cores) also has to be used for packet forwarding for BCube and DCell. Since BCube and DCell are server-centric architectures with fewer switches than switch-centric

topologies servers often have to perform the packet forwarding. For BCube, however, switches take on a greater share of the job of packet forwarding than servers do compared to DCell.

Furthermore, it can also be observed that for the random VSCP the DCell performance decreases stronger than that of BCube. This is because the traffic load in DCell is more imbalanced than BCube: The level-0 links carry much higher traffic than the other links. As a result, the aggregate throughput of DCell is smaller than that of BCube.

4.6.3.2 Impact of VSCP Strategies

An example result is shown in Figure 4.10 for the successful embeddings and in Figure 4.11 for the relation cost-to-embedding in the case of the Fat-Tree topology. The x-axis indicates the number of servers in the DC topology and the y-axis the number of successfully embedded VNF chains. To test the performance of the vendor-based VSCP the assumption is that each VNF on the VNF service chain is from a different vendor and has to be placed on a different server. The performance of the vendor-based VSCP strategy is closer to the local strategy as the VNF functions are often placed locally close, too but on different servers. Further, 2-/3-tier, BCube and DCell using different VSCP strategies were also simulated. The general result trends are similar to Fat-Tree.

For the random VSCP strategy the path within a DC tends to be longer than the local VSCP for all DC topologies, which also results into more consumed bandwidth and fewer embedded VNF service chains. The difference between these two strategies gets greater once the size of a DC becomes larger, as for larger DC size the hop count between the VMs increases.

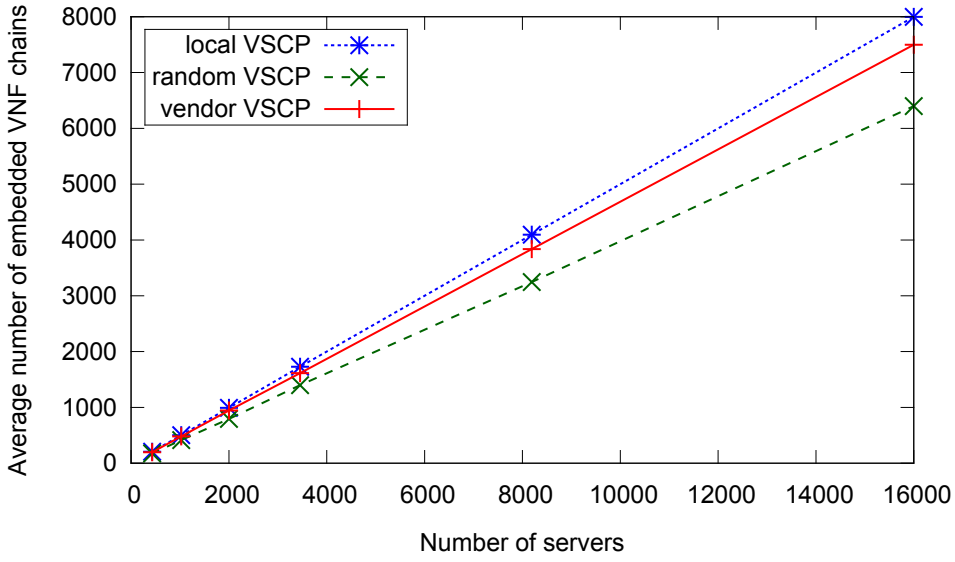

Figure 4.10: Impact of the different VSCP strategies for Fat-Tree (with 5 Gbps chain capacity) [5]

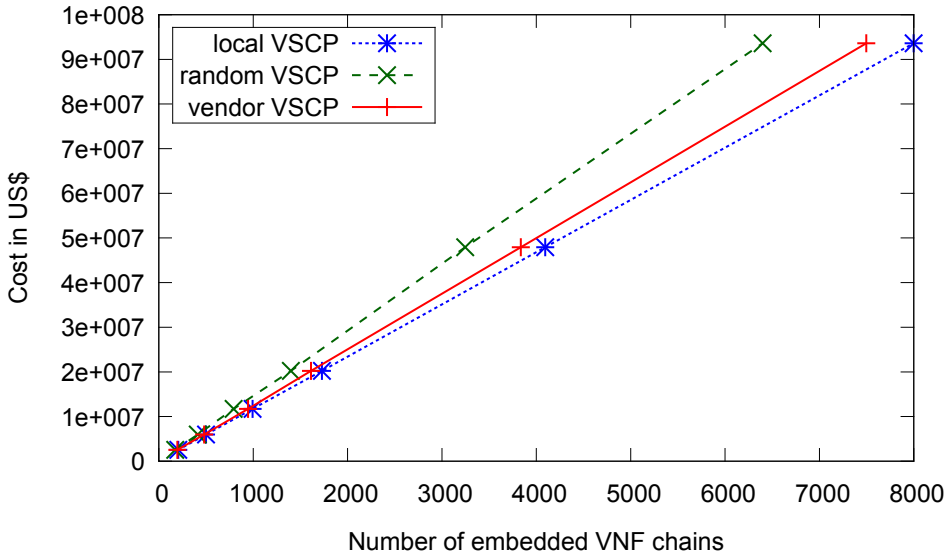

Figure 4.11: Cost of the different VSCP strategies for Fat-Tree (with 5 Gbps chain capacity)

4.6.3.3 Impact of VNF Service Chain Length

In Figure 4.12 the amount of embedded VNF service chains is depicted for the VNF chains with different length for DC with Fat-Tree topology. In general, a longer VNF service chain often requires more computing and bandwidth resource from a DC.

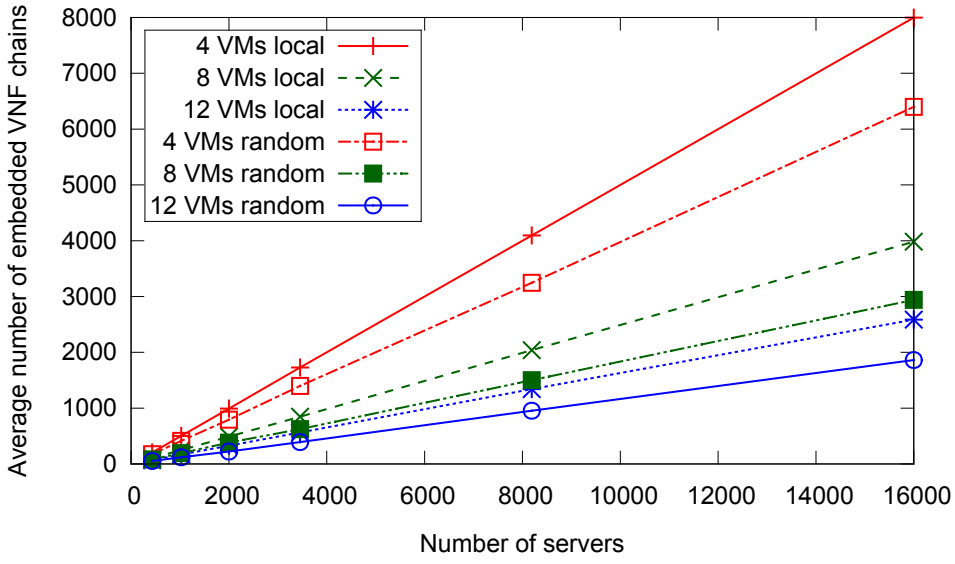

Figure 4.12: Impact of the VNF service chain length for Fat-Tree with 4, 8 and 12 VMs per service chain (with 5 Gbps chain capacity) [5]

Here the performance of an 8 VMs VNF chain is compared with a 4 VMs one for the local VSCP, and the observation is that the number of successfully embedded VNF chains is about half of the one with short VNF chain. For the random VSCP, the number of successfully embedded VNF chains is less than half of the one with short VNF chain, which means that random

VSCP has more impact on the VNF chain length. For instance for a DC with size 16000 servers, using random VSCP has 20% performance drop in terms of the number of embedded chains with chain length 4 VMs, 26% drop for the chains with 8 VMs and 28% for the chains with 12 VMs. The reason is that for the random VSCP with longer chain (e.g. 8 VMs) the embedding path lengths get about twice as long as for the short 4 VMs chain.

4.6.3.4 Impact of Traffic Load

The impact of different traffic input into the VNF service chains is examined in this part to see if there is some performance advantage using smaller input traffic loads. The VNF chain processing capability is varied from 1 Gbps to 5 Gbps. Compared to 5 Gbps input traffic load the average hop count between two VMs in the VNF chain is lower. If two VMs are on the same server the hop count is zero. The total traffic load (i.e. sum of the input load of each embedded VNF chains) is about the same for VNF service chains with 1 Gbps and 5 Gbps input traffic.

Furthermore, the VSCP strategy impact on the maximum number of VNF service chain embedding with different VNF service chain traffic load is investigated. The results are shown in Figure 4.13. To be able to compare all DC topologies together the number of servers is fixed at around 3500. For all topologies the random VSCP can embed fewer VNF chains than the local VSCP. This is because when the random VSCP is used the aggregation links become the bottleneck of 3-tier architecture. For DCell, the bottleneck are the links from server to switch. However, such difference is very small (almost no difference) for Fat-Tree and 2-tier when the VNF service chain capacity has a low value, i.e. 1 Gbps.

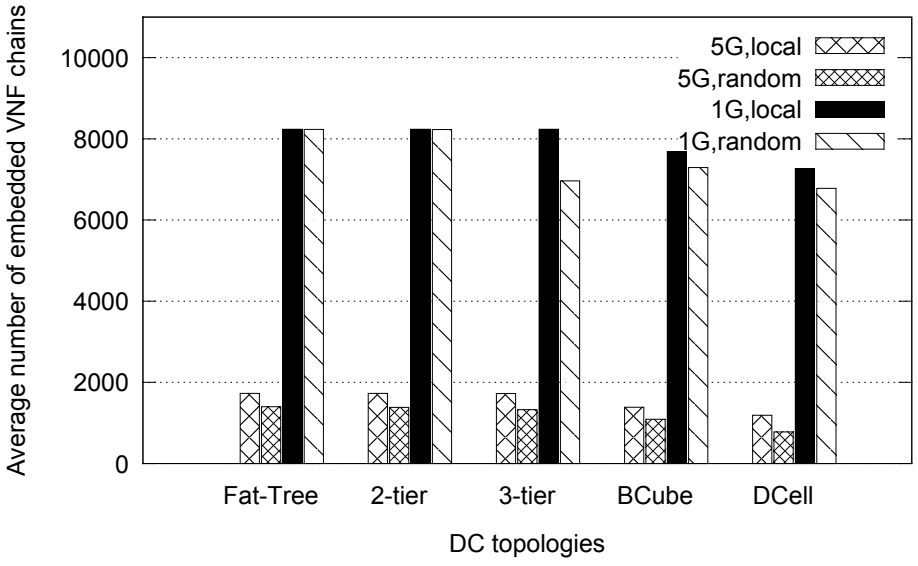

Figure 4.13: Impact of the VNF service chain traffic load with chain length 4 VMs and about 3500 servers [5]

Once the VNF service chain capacity is increased, meaning there will be less VNF embedding requests, the Fat-Tree and 2-tier also show performance difference between random and local VSCP. The reason is that for higher input traffic also more CPU cores are required and for random VSCP it gets difficult to find servers with enough unoccupied cores after several embedded

VNF chains. Therefore, the CPU resources cannot be fully utilized and as a result fewer VNF chains were embedded.

The Fat-Tree topology is investigated using the local VSCP strategy. Using 1 Gbps input for the VNF chain results in not exactly 5 times more VNF chains as for the 5 Gbps input, which is about 2 to 5% lower. However, the bandwidth usage in the DC is much lower (about 60% lower). The reason is that with 1 Gbps input a 4 VMs service chain can always be embedded on an unoccupied server, which does not require inter-DC packets forwarding. However, this is not the case for 5 Gbps traffic load, which requires more cores as an unoccupied server can offer.

4.6.3.5 Impact of the Number of Servers per Rack

In addition, different numbers of servers (in this case 24, 48 and 96) per rack are compared. Here, the 2-tier and 3-tier architectures are taken as the scenario for discussion. Increasing the rack size for 2-tier architecture will result in a larger DC size, which means that the rack size has a high influence on its corresponding DC size. Four-core switches are used for the scenario with 24 and 48 servers per rack and eight-core switches are used for the scenario with 96 servers per rack.

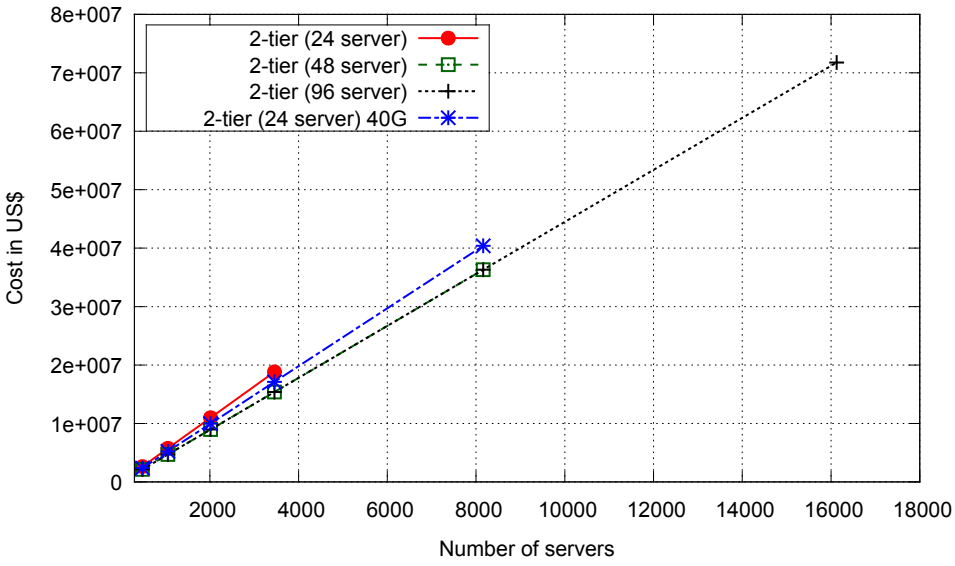

Figure 4.14: Cost of the 2-tier architecture with different numbers of servers per rack [5]

For 96 servers per rack the DC can at least scale up to 16000 servers. Further, also an alternative 2-tier architecture with 40 GbE links between ToR and 8 core switches is considered. For the other 2-tier configurations this is not possible due to the limited number of server ports at the switches. The performance is similar to the 2-tier with 24 servers for the local VSCP and random VSCP. The larger rack size architecture has embedded about 3% less VNF chains. However, the cost for the larger rack size 2-tier architectures is a bit lower using larger rack sizes as shown in Figure 4.14. Comparing the 2-tier architecture with 96 servers per rack to the Fat-Tree it can be observed that it has a slightly lower cost than Fat-Tree.

Simulations with different rack sizes have also been done for the 3-tier architecture. The overall cost can be reduced by using larger rack size. However, the 3-tier architecture is more influenced by the rack size. The larger the rack size the more bandwidth is required in the aggregation links to keep the over-subscription rate low. With larger rack size avoiding high over-subscription rate becomes more difficult; consequently, the performance decreased and fewer numbers of VNF service chains could be embedded, especially for long chains.

4.6.3.6 Evaluation Results and Discussion

This section summarizes the DC impact factors to deploy high traffic volume NFV type applications. The 2-tier architecture performs well for VNF chains embedding with a corresponding low cost compared to the other DC topologies. However, the number of ports per switch limits its scalability since every core switch is connected to every ToR switch forming a complete mesh network. For a large DC size switches with high port numbers are required; however, this does not always exist (e.g. 100 GbE switches are only available with up to 192 ports currently). Scaling up the rack size in the 2-tier topology can result in decreasing the cost; however, this also ends in an additional scalability problem for the low-cost ToR switches.

The 3-tier architecture also indicates suitable cost performance for VNF service chain embedding; however, it is not as robust with VSCP strategies as the random placement. The 3-tier architecture has a bandwidth bottleneck at the aggregation layer, even with no over-subscription at this layer. Comparing different over-subscription rates for 3-tier architecture it demonstrates that the over-subscription rate plays an important role in the performance of the VNF chain embedding, i.e. low over-subscription rate results in a good performance but high cost. In contrast, the higher the over-subscription rate, the lower the cost, which also results in the lower amount of embedded VNF service chains. With no or close to no over-subscription the performance of the 3-tier is similar to the 2-tier architecture; however, the scalability problem due to the limited number of ports of the switches is relaxed compared to 2-tier.

Fat-Tree overcomes the bottleneck of the conventional tree (2-/3-tier) by introducing more bandwidth into the switches near the root. It is easy to scale up for large DCs due to the low switch port number. The Fat-Tree performs well in terms of robustness when different VSCP strategies are applied for VNF chains embedding. However, it has the highest cost compared to the other DC topologies with the same number of servers because of the high amount of required switches.

Compared to Fat-Tree and 2-tier architecture the performance of BCube is lower. The reason is the additional computing resource needed at the servers to execute packets forwarding instead of using switches for forwarding. It has a fully meshed architecture, which makes the cabling more difficult compared to the tree-based architecture (2-/3-tier and Fat-Tree).

The performance of DCell is even lower than BCube. The reason is that, compared to BCube, more forwarding is done at server level for DCell, which also means that the links between servers and switches are also easily overloaded. Further, the fully meshed structure makes the cabling even more difficult compared to the BCube architecture, which makes DCell hard to be used for larger DCs. Also, it is not straightforward to add any number of servers in the DC because of the double-exponential growth of the servers in the network. With DCell scaling

doubly exponential with the server node degree, adding a small number of servers can create an imbalanced and not fault-tolerant structure.

4.6.3.7 Recommendations for Operators

From the evaluation results it can be observed that the placement strategy plays an important role for the NFV type application deployment according to different vendors' policy and preferences. However, a significant impact in terms of deployment cost cannot be identified. For lower input traffic (as, e.g. 1 Gbps per service chain) the random and local VSCP perform similarly – delivering nearly the same number of embedded VNF chains as with short chain length.

The DC topologies can be evaluated from many aspects in terms of VNF service chain deployment. For instance, the architecture that can support high resource utilization or lower cost to construct such DC is preferred. Due to scalability issues it is not straightforward to compare deployment cost for NFV type applications for all type of DCs. The scenarios for small-scale DCs (less than 4000 servers) and for large-scale DCs (more than 8000 servers) have to be considered separately.

For small DCs, the 2-tier architecture performs well – while the Fat-Tree architecture performs well for large DCs and can scale to very large server numbers. Therefore, the 2-tier architecture is the 'best' choice for small-scale DCs for hosting NFV applications. For large DCs the Fat-Tree architecture is the 'best' and can scale to large server sizes over a hundred thousand servers using low-cost switches.

4.6.4 Cost versus Availability for VNF Service Chains in DC Networks

In the following a parameter study is done to evaluate which parameters have great influence on the performance and how the different topologies behave to the changing parameters. The same simulation framework as in the last section is used for analyzing the cost and availability performance of the different DC topologies for the deployment of resilient VNF chains. However, the algorithm for achieving high availability is added to the framework.

The simulation settings are extended to the availability parameters for the DC in Table 4.3 and the VNF service chain. The assumption for the traffic load is that the maximum traffic load that can be processed by a VNF service chain is 1 Gbps. For each parameter the simulation is run 100 times and the average number of successfully embedded VNF chains is calculated.

4.6.4.1 Different DC Sizes

The first simulations evaluate the influence the different DC sizes for each topology. The DC size is varied between 400 and about 4000 servers and the local VSCP strategy with the backup strategy 1 is used.

The result of the performance of the different DC topologies for a requested service availability of ‘five nines’ can be seen in Figure 4.15:

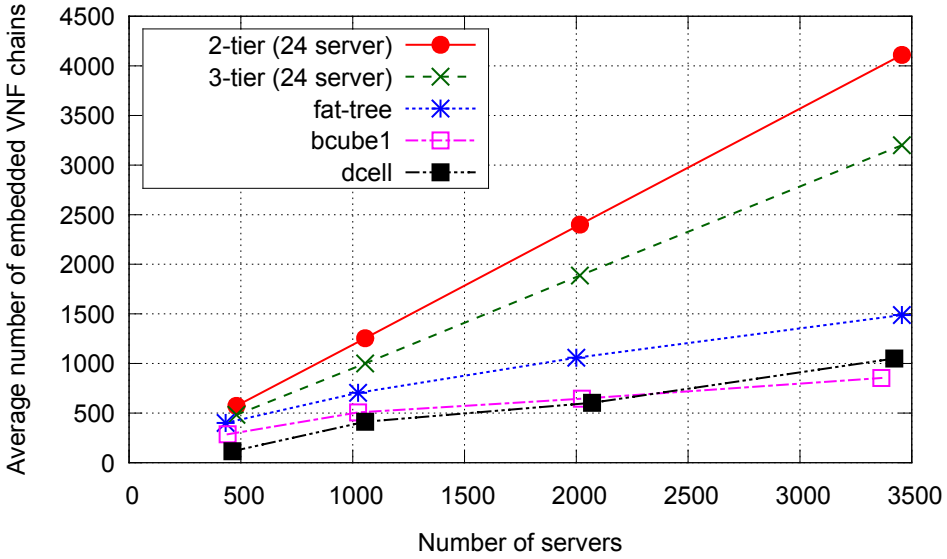

Figure 4.15: Successfully embedded VNF chains for service availability of 0.99999 and local VSCP strategy [6]

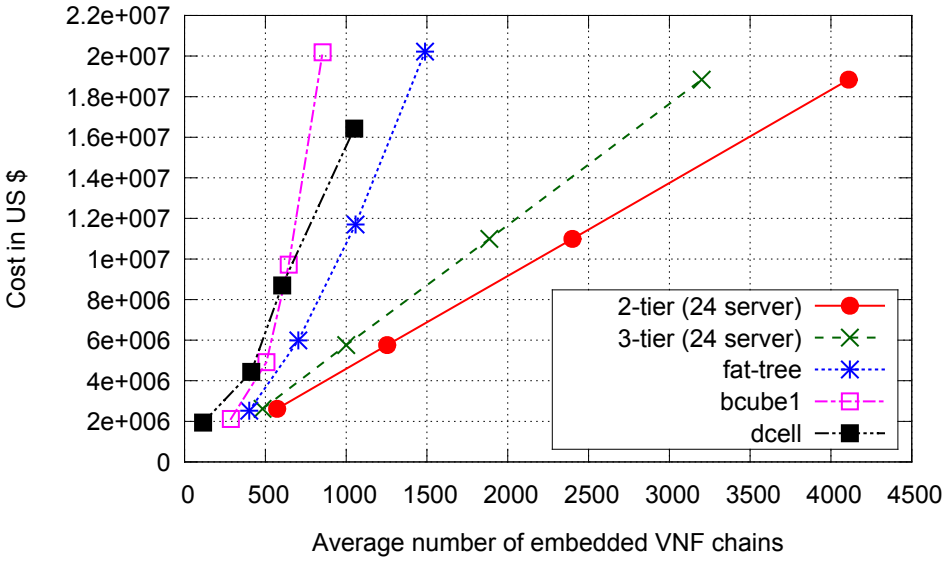

Figure 4.16: Relation cost vs. successfully embedded VNF chains for service availability of 0.99999 and local VSCP strategy [6]

In this scenario the 2-tier topology performs best. The second best in this case is the 3-tier topology. One reason for this performance is the fact that these two topologies use modular core switches with high availabilities. The other topologies were mostly built with low-cost switches with lower availabilities. Furthermore, the results show that the Fat-Tree topology has a higher embedding rate compared to the BCube/DCell topologies. This can be attributed to the fact that Fat-Tree contains more switches in its DC topology whereas BCube and DCell are server-centric DC topologies. BCube and DCell partly use servers to work as switching nodes,

which result in less successful VNF chain embeddings due to the lower availability of servers compared to switches.

Figure 4.16 presents the corresponding cost in relation to the number of average embedded VNF service chains. For comparing the cost the model of server and switch cost from Section 4.4 is used. Here, too the 2-tier architecture has the lowest cost in all cases.

4.6.4.2 Different requested Service Availability

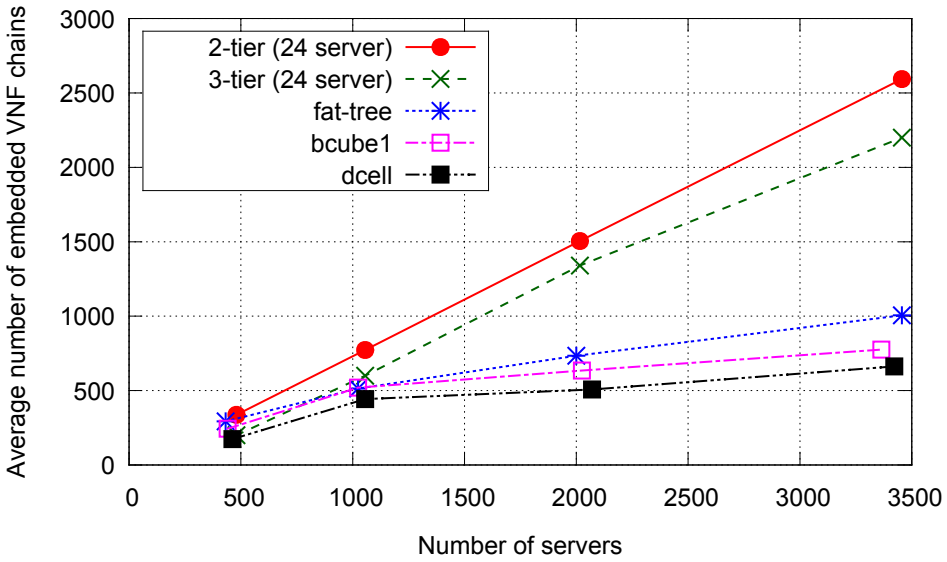

Figure 4.17: Successfully embedded VNF chains for service availability of 0.999999 and local VSCP strategy [6]

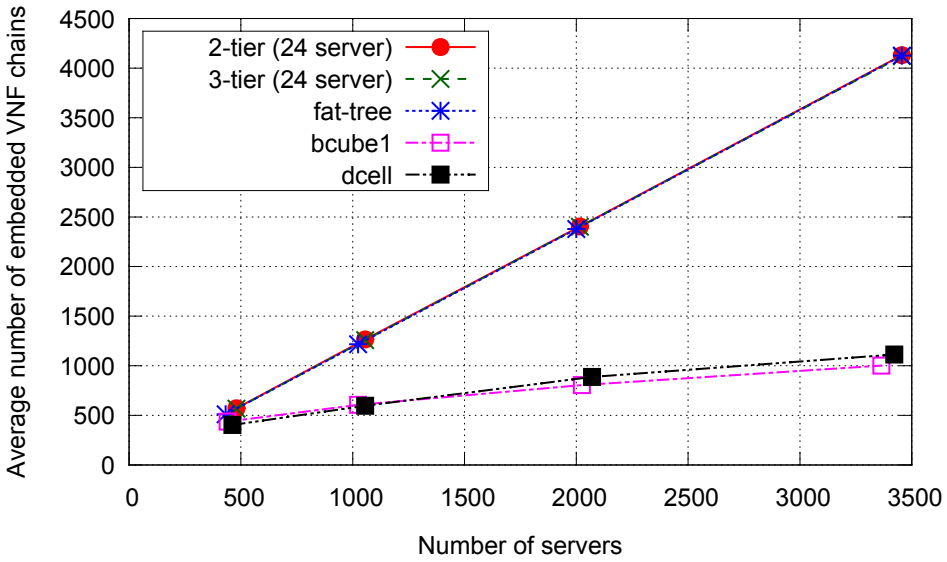

Figure 4.18: Successfully embedded VNF chains for service availability of 0.999 and local VSCP strategy [6]

Next, the DC topologies are examined with different requested service availability levels from ‘three nines’ to ‘six nines’. With rising requested service availability the number of successfully embedded VNF chains decreases: Especially Fat-Tree shows more successful embeddings for ‘three nines’ requested service availability (Figure 4.18) than for ‘six nines’ (Figure 4.17).

For ‘six nines’, each VNF service chain needs two backup chains on average to achieve that high service availability. With the highest availability requirement (i.e. six nines) the average number of embedded VNF is the relatively lowest one due to the fact that it becomes difficult to embed the backup chains in a (fully) disjoint way. In addition, the longer the chains become the more elements in the chain and the lower the availability will be. When the service availability decreases one single backup chain is sufficient in all cases. While the absolute number of chains that can be embedded decreases for all topologies, which is to be expected, their relative ranking does not change.

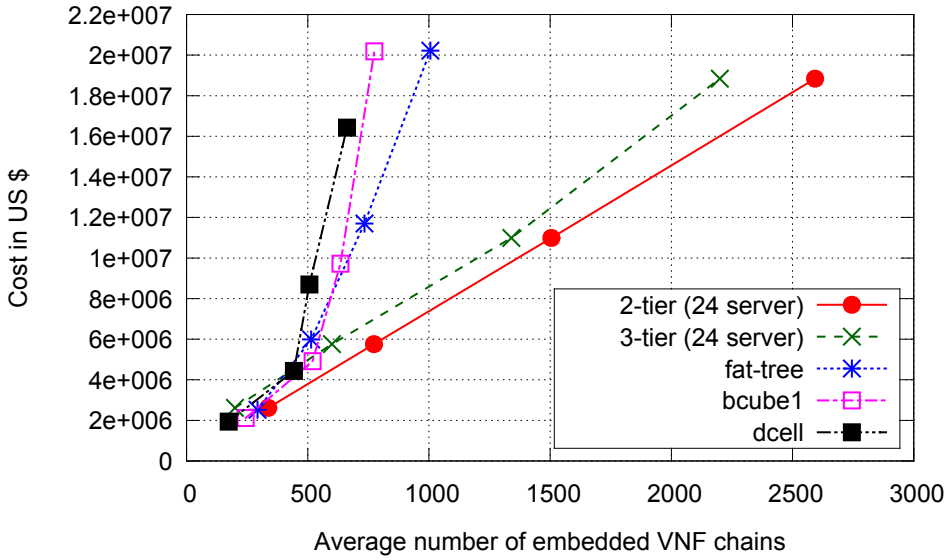

Figure 4.19: Relation cost vs. successfully embedded VNF chains for 0.999999 and local VSCP strategy

Figure 4.19 and Figure 4.20 show the corresponding cost in relation to the average embedded VNF service chains for service availability of ‘three nines’ and ‘six nines’.

In Figure 4.21 different requested availabilities and the successfully embedded service chains are shown for the 2-tier topology. For ‘three nines’ to ‘five nines’ with the 2-tier topology the resulting number of successfully embedded VNF chains are nearly identical due to the reason that the embedding behavior is the same. On average, the same number of backup chains are needed to be embedded successfully (i.e. the number of backup chains achieve an availability of 0.99999; however, removing one backup chain not even an availability of 0.999 can be achieved).

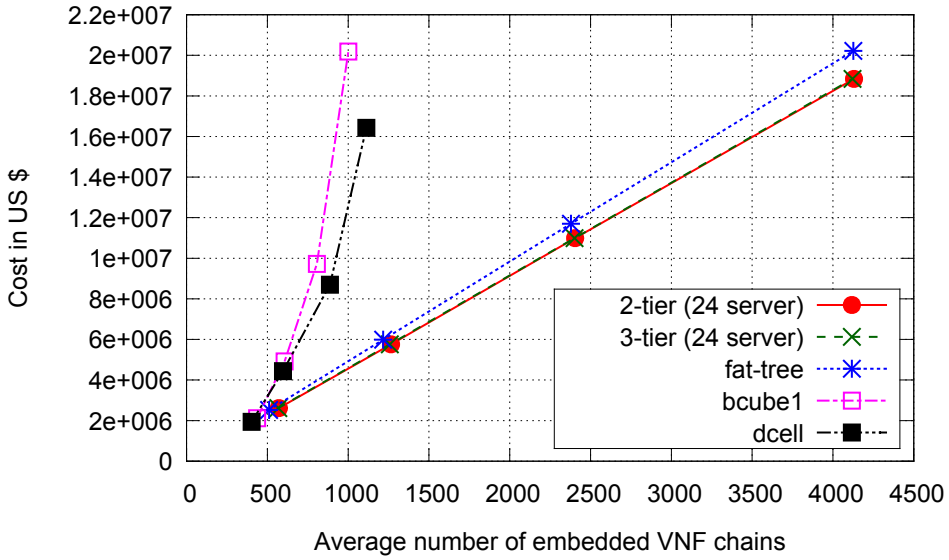

Figure 4.20: Relation cost vs. successfully embedded VNF chains for 0.999 and local VSCP

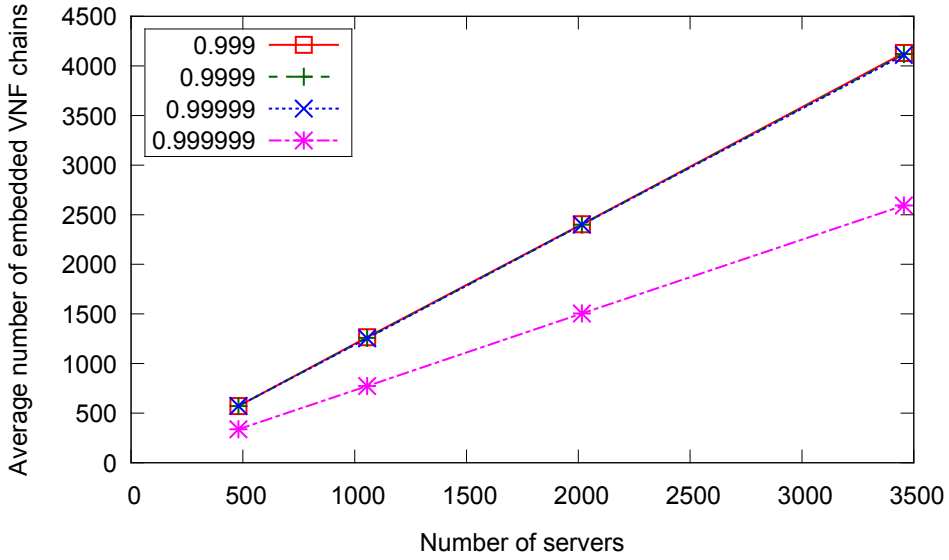

Figure 4.21: Impact of the different requested service availabilities for 2-tier [6]

4.6.4.3 Different VSCP Strategies

Furthermore, the local and vendor-based VNF service chain placement algorithms for the backup deployment strategy 1 are compared. The assumption for the VNF vendor-based VSCP strategy is that each function of the service chain (being from a different vendor) has to be placed on a different server. In this case the routing paths for primary and backup chains within a DC tend to be longer than for the local VSCP at all DC topologies.

Further, often an additional backup chain is required for achieving the same availability if the vendor-based VSCP is used. The result is higher bandwidth and VM consumption and therefore, fewer VNF chains can be embedded. For example, with a requested availability of ‘five

nines' the 2-tier topology with the local VSCP can embed about double the number of VNF chains as in the case of using the vendor-based VSCP that needs an additional backup chain (see Figure 4.22). The same effect can be experienced with higher availability values. If the requested service availability decreases, the influence of the VSCP strategies is decreased in the embedding and the resulting embedded VNF chains.

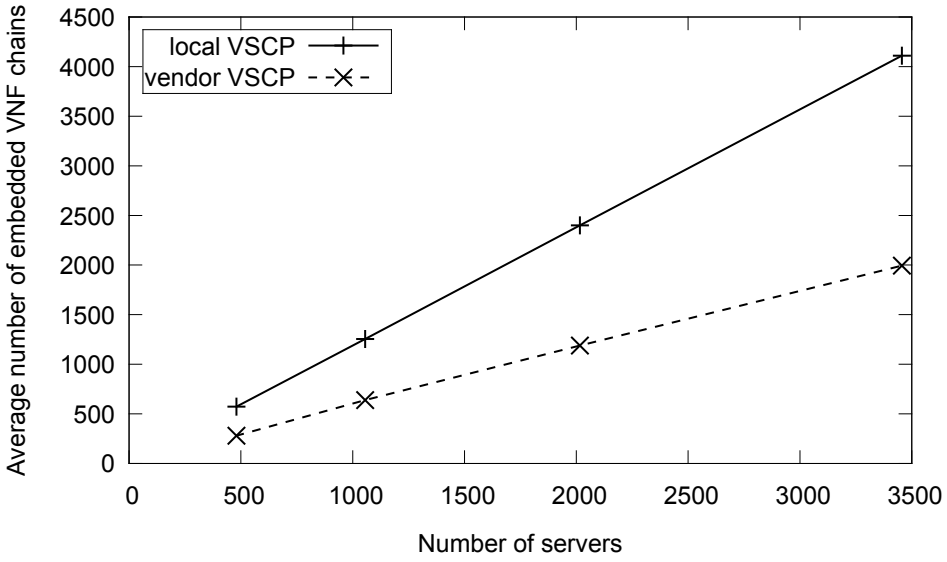

Figure 4.22: Impact of the different VSCP strategies on the example of for 2-tier topology

An example result for the different DC topologies with a server size of about 3500 is illustrated in Figure 4.23 for different VSCPs and availabilities.

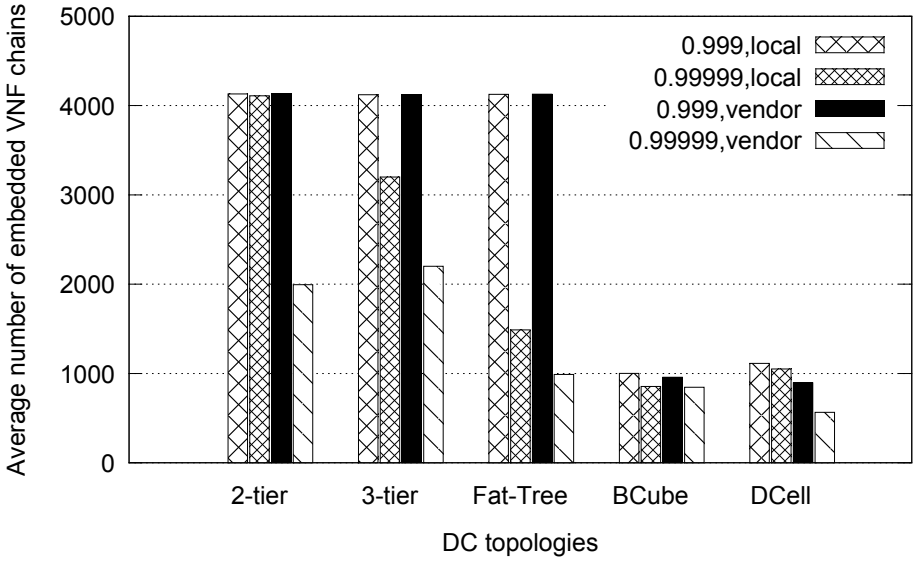

Figure 4.23: Impact of the different VSCP strategies for requested service availabilities 0.99999 and 0.999 with about 3500 servers [6]

However, in Figure 4.23 it can be recognized that for a service availability of 0.999 the VSCP strategy has no or very little influence on the embedding for all topologies. Consequently, the number of embedded chains is close to identical for this case.

4.6.4.4 Different Backup Deployment Strategies

The different backup deployment strategies are also compared. For the simulation the deployment strategy 2 (resource pooling) with the vendor-based VSCP strategy is examined. If strategy 2 is combined with the local VSCP it would be equal to strategy 1 with the local VSCP.

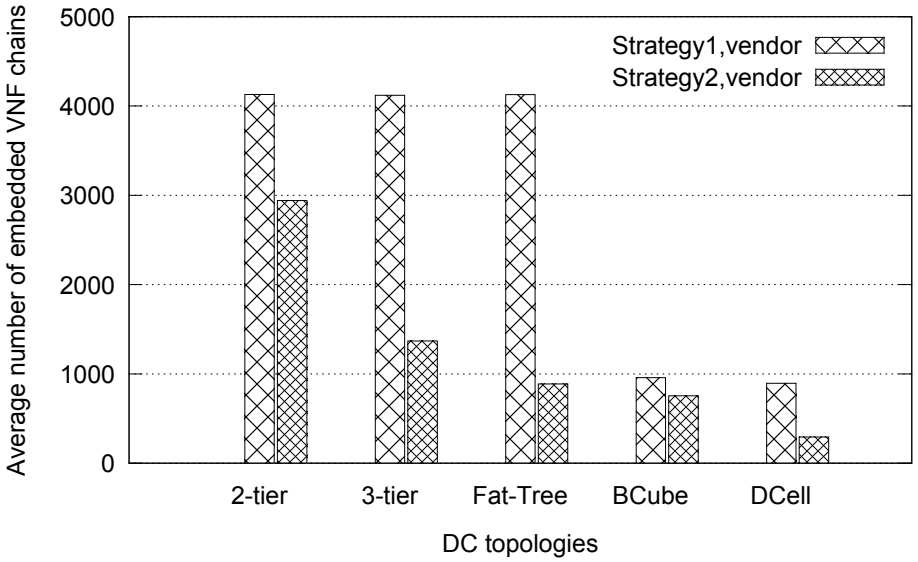

Figure 4.24: Impact of the backup deployment strategy for availability 0.999 with about 3500 servers [6]

In Figure 4.24 strategy 1 with vendor-based VSCP is compared to strategy 2 for the different DC architectures with about 3500 servers and significant decrease in the performance of VNF service chain embedding can be observed. From the simulation, strategy 2 (resource pooling) can only successfully embed the VNF chains with the requested availability for 'three nines' and 'four nines'. This can be attributed to the common LBs in the backup graph, which become critical and prohibit high availability values. For a requested availability of 0.9999, Fat-Tree, BCube, DCell cannot successfully embed the chain. This is due to the facts that the LB is embedded in one of the switches. However, the switches in Fat-Tree, BCube and DCell have lower availability than those switches (especially the core switches) used for the 2-/3-tier topologies. Embedding the LB in one of the high-available core switches the 2- or 3-tier topology can result in a higher availability VNF chain leading to more successful VNF chain embeddings. Also, the very low VNF chain embedding number of DCell compared to the other topologies as shown in Figure 4.25 results from the fact that DCell has much fewer switches than the other DC topologies and consequently can hardly successfully embed all the LBs in strategy 2.

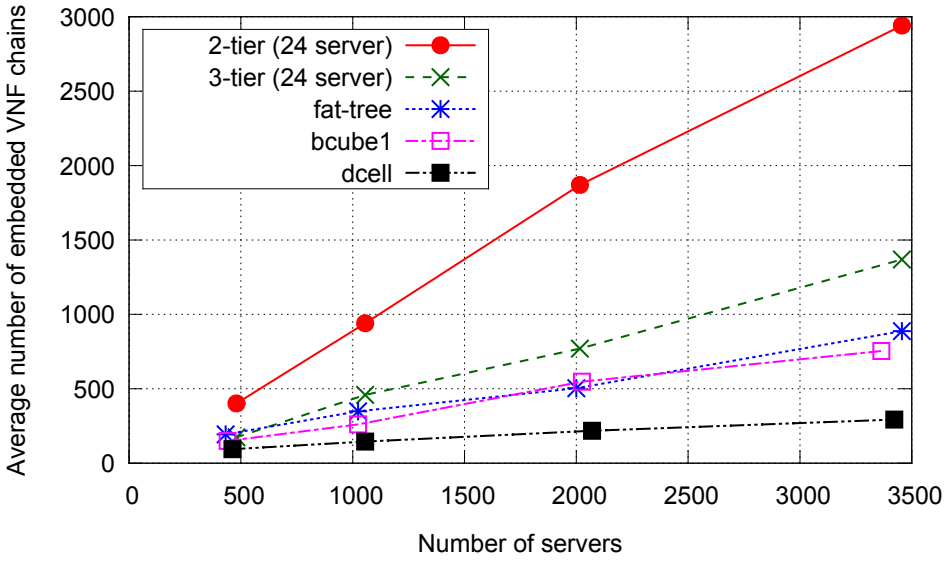

Figure 4.25: Successfully embedded VNF chains for backup deployment strategy 2 for availability 0.999

4.6.4.5 Evaluation Results and Recommendations for Network Operators

Finally, the overall performance of the different DC topologies is compared to their cost and the ability of high availability embedding. The results from the number of successfully embedded service chains and the cost for a DC with defined server size in the range between 400 and 4000 servers show that the 2-tier has the lowest cost in relation to the performance and the most reliable embedded VNF service chains.

In summary, the switch-centric topologies are better suited for achieving high availability values. The 2-tier architecture shows the best performance followed by the 3-tier topology. The fully meshed of the 2-tier architectures has advantages in embedding against the 3-tier architecture as there are more paths to route backup chains. Generally, high-cost and reliable switches are needed in the DC topology to achieve high availability service chains. The server-centric topologies have the disadvantage that the servers are less reliable than switches and, therefore, the performance of these topologies for reliable VNF is lower as compared to the switch-centric ones.

Further, inefficiencies arise when employing a vendor-based VNF service chain placement (VSCP): Due to the fact that an individual server must only contain VMs from a single vendor, the primary and backup chains generally get longer compared to a local VSCP and additional backup chains are required so that in the end fewer chains can be embedded.

The recommendations for network operators derived from the results are that switch-centric topologies are better suited when using low-cost devices (especially low-cost servers) since the switches have higher availability than the servers. Therefore, the 2-tier topologies can be recommended for achieving high availability for services; however, the scalability is limited. Therefore, for large DCs it is less suited. The combination of the scalability of the Fat-Tree with high-cost switches could be a solution for large DCs for hosting services requiring high

availability at a reasonable price. Since the scalability of the Fat-Tree is much higher than 2-/3-tier tree topologies and topology consists also three levels of switches like 3-tier, the overall performance while using high-cost switches could be similar or better than 3-tier.

4.6.5 Evaluation Summary

Finally, the two evaluation blocks of VNF service chain embedding versus performance and availability are combined and results are derived. In summary, derived from the simulation results, suitable topologies for the VNF service chains are 2-tier architecture and Fat-Tree architecture. These two topologies perform well for embedding the VNF service chains while also resulting in lower cost compared to the other topologies. However, with additionally considering high availability for the VNF service chain the 2-tier architectures is a better choice than the Fat-Tree. Even the 2-tier topology is well suited for VNF service chains requiring very high service availability. 2-tier architecture lacks scalability and therefore, is less recommended for large DCs with several ten thousands or hundreds of thousands of servers. In case of large or mega-DCs, 3-tier (if it can scale up to the requested server size) or Fat-Tree could be recommended for deploying highly available VNF service chains.

Furthermore, the placement strategy is important for NFV type application deployment according to different vendors' policies and preferences. Especially on the service availability the placement strategy has a large impact. From the simulation results it is obvious that using network functions from different vendors can decrease the efficiency in the DC. Therefore, it has to be considered if the cost could be lower to use VNFs of one vendor or a few different vendors.

4.7 Chapter Summary

This chapter addressed the analysis of cost versus availability for VNF service chains in DC networks. The different parts of the network environment, DC network and VNF service chain were introduced and modeled.

An embedding algorithm was developed to achieve a requested service availability value for each the VNF service chain in the DC network. Furthermore, the ability of different DC architectures to deploy resilient NFV type applications using this algorithm was examined. With intensive simulations the different DC topologies were evaluated for the use of VNF service chains and their cost as well as to achieve high availability of the service chains. Different backup deployment strategies and VNF service chain placement strategies were compared for each DC topology. The simulation results were evaluated and further discussed to give recommendations for the (mobile) network operators for the best suitable DC topology. From the results the 'best' DC topology for achieving high availability at the lowest cost for VNF service chain is a 2-tier tree topology. Another suited topology with a good scalability for the DC is the Fat-Tree topology; however, lower service availability can be achieved for the VNF chains.

5 Conclusion and Future Work

5.1 Conclusion

The communications industry is currently undergoing a shift – following the Web/Internet world away from component reliability towards system reliability, i.e. building reliable end-to-end (E2E) networks based on less reliable sub-components. While this paradigm has been largely successful in the Web/Internet world, its benefits have not yet been proven unambiguously in the telecommunications industry. The approaches presented in this thesis take a first – although limited for two different networks – step in formalizing such analysis and shedding some light onto the debate.

This thesis presented a detailed study of the trade-off between cost and availability in different networks. Two example networks, optical fiber wide area network and data-center (DC) network, were chosen to show the trade-off. Based on these two networks conclusions on the trade-off between cost and availability were found.

First in Chapter 2 the fundamentals of network virtualization, optical transport networks and DC networks, reliability and availability were presented. Furthermore, a study of the failure characteristics of IP networks in general and especially of optical fiber networks and DC networks was done. In addition, a detailed introduction in the topic of virtual network embedding (VNE) and the algorithms for VNE with a focus on survivability and reliability were given.

In Chapter 3 the trade-off between cost and availability in virtual networks for optical fiber transport networks was investigated. For this study the problem was formulated as a VNE problem with explicitly considering availability requirements for the links. An embedding algorithm was developed to achieve a high availability for the virtual networks on top of the physical network and as a special case for the fiber transport network. The developed heuristic solves the problem in polynomial time. The idea of the algorithm is to calculate the primary paths and if needed one to several backup paths which together have the requested availability. As the physical network links cannot always provide the requested availability several independent parallel links or paths are combined to achieve the availability.
To minimize cost network operators need to consider the required network availability already at the network design stage. Two different approaches were considered in the thesis. One end of the design space is marked by the ‘low-cost physical’ approach whereas as little money as possible is spent on physical protection. Instead, high availability is realized by combining multiple parallel paths to form one virtual path or link. The other end of the design space can be described as a ‘high-cost physical’ approach. Here, enough money is spent on the physical

network to allow already single paths to achieve a requested availability level. To examine the underlying trade-off between these two philosophies a model that sets the network deployment cost in fiber networks in relation to the achieved resiliency was defined and a cost function to determine the overall cost when realizing a virtual network with a requested availability on a physical network with a different availability was created. With the combination of this cost function and the algorithm the minimum embedding cost for a VNE can be calculated.
In addition to modeling cost with regard to MTBF, the impact of different parameters including cost scaling factor, network size and network topology on network embedding cost was studied. The theoretical framework was applied to a number of different network topologies – both artificial grid topologies and real-world, existing country- and continent-wide networks. With intensive simulations evaluating the trade-off between cost and component availability when realizing virtual networks, several results have been found. The results show that for most configurations the 'low-cost physical' approach with low or medium reliability levels in the physical network results in the lowest cost. This is especially interesting as these reliability levels fit very well to parameters from real fiber deployments: already the lowest availability values for buried fiber found in the literature are sufficient. Therefore, it seems advisable to realize availability in the virtual domain rather than in the physical domain in networks.

Chapter 4 presented the analysis of cost versus availability for virtualized network function (VNF) service chains in DC networks. The different parts of the network environment, DC network and VNF service chain were introduced and modeled. An embedding algorithm was developed to achieve a defined reliability in the VNF service chain and especially the requested service availability value.
Intensive simulations were done to evaluate the different DC topologies for the use of VNF service chains and their cost and also to achieve high availability of the service chains. The simulation results were evaluated and further discussed to give recommendations for the (mobile) network operators for the best suitable DC topology. Derived from the simulation results suitable topologies for the VNF service chains are 2-tier architecture and Fat-Tree which perform well for embedding the VNF service chains while also requiring lower cost compared to the other topologies. Fat-Tree topology is built with only low-cost commodity components (switches and servers). Further, the 2-tier tree topology uses only a limited number of high-cost modular switches in addition to the low-cost servers and ToR switches. It was shown that with low-cost components it is possible to gain suitable performance for VNF service chains and reach a defined availability while embedding in DCs. However, with also considering high availability for the VNF service chain the 2-tier architectures is a better choice than the Fat-Tree, especially for small DC sizes.
From the result it can be concluded that using low-cost devices (e.g. low-cost switches and servers) in DC networks high available networks for telecommunication are possible to build. However, for very high available networks often the approach of using dedicated specialized switches in combination with low-cost switches and servers still achieves lower cost.

In conclusion from these two example trade-off studies it can be observed that using low-cost devices and network components as well as virtualization technologies, it is possible to achieve a high service availability while saving cost in comparison to highly costly components. Therefore, it seems advisable to realize availability in the virtual domain with special protection mechanisms rather than in the physical domain based on dedicated hardware or hardware protection mechanisms. However, for services requesting very high availability (i.e. more than five

nines) often the approach of using dedicated specialized devices or hardware in combination with low-cost devices in the networks still achieves the lowest cost.

5.2 Future Work

This thesis studied various important problems in the area of reliability and availability in virtual network embedding and network virtualization with the trade-off between cost and availability. Overall, the findings and conclusions from this effort open up new avenues for future research, some of which are now highlighted.
Here an outlook is given for future work which can extend or reuse the work in previous chapters. The following fields could be future research activities.

Extend the cost model to consider OPEX of the different networks In the study in this thesis, only CAPEX is considered for the cost model for the fiber and DC networks. The cost versus availability study could be extended to also consider the OPEX for the different networks. A combined CAPEX and OPEX analysis could be done to verify if different behavior arises.

Consider multi-provider environment The presented evaluations in the fiber and DC network consider a single provider environment. By combining the two different networks, the wide area transport network and the DC network, a complete high-available E2E telecommunication network can be created. However, owning the complete E2E (telecommunication) network or renting the network from one single PIP could result in high cost for the operator. A multi-provider environment could be considered where different PIPs having different infrastructure with different availability/reliability and offering their physical resources for different cost. The trade-off study then could be extended to find the cheapest PIP for the requested E2E virtual network and the requested service availability.

Examination of additional DC topologies or design of a specialized DC topology for NFV application Further, other additional DC topologies could be examined for the use of NFV applications. It could be investigated if these other DC topologies are better suited in case of small or large DC networks or even both compared the the selected topologies in this thesis in Chapter 4. Example DC topologies could be MDCube, Hyper-BCube or FiConn, which are essentially an extension of the BCube and DCell architecture. MDCube [134] is a high-performance interconnection structure to scale BCube-based containers to mega-DCs. Hyper-BCube [135] combines the advantages of both, DCell and BCube architectures while avoiding their limitations. Hyper-BCube strikes a compromise between the excessive scalability of DCell and high cost of BCube. FiConn [136] is another example of a recursive structure and server-centric architecture. FiConn utilizes both ports of the commodity server machines actively for the network connections and tries to build a low-cost interconnection structure without the expensive higher-level large switches. Compared to Fat-Tree, the number of used switches is smaller in FiConn and compared to DCell, the wiring cost is less. Also, the combination of different DC topologies (e.g. the advantages of each) could result in a design of a new DC topologies specialized for the VNF services with high service availability requirements.

Extend the trade-off study of cost versus availability with a mathematical theoretical analysis The investigation in this thesis is mostly based on simulation, especially Chapter 3. In addi-

tion, the algorithm used for solving the problem is a heuristic whose performance with respect to the optimum or a bound is not completely provided. Therefore a mathematical theoretical analysis on how external factors affect the cost could be done. The results of the theoretical analysis could be compared to the simulations based one in this thesis.

Improvement of the heuristics protection algorithms First, further improvements to the heuristic strategies can try to close the performance gap with an optimal solution through additional resource savings. A key objective here can be to minimize further backup resource usages at the link protection in the fiber network or the VNF service chain backup for the DC network. Some other possible strategies can also include further improved resource sharing in the protection methods, which still satisfies the availability constraint.

Bibliography

[1] Sandra Herker, Ishan Vaishnavi, Ashiq Khan, and Wolfgang Kellerer. Use Cases and Derived Requirements for a Reconfigurable Mobile Network. In *IEEE International Conference on Communications Workshop on Advanced Mobile Networks (ICC Workshop)*, pages 5503–5507, Ottawa, June 2012.

[2] Sandra Herker, Ashiq Khan, and Xueli An. Survey on Survivable Virtual Network Embedding Problem and Solutions. In *ICNS 2013, The Ninth International Conference on Networking and Services*, pages 99–104, March 2013.

[3] Sandra Herker, Xueli An, Wolfgang Kiess, and Andreas Kirstädter. Path Protection with Explicit Availability Constraints for Virtual Network Embedding. In *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pages 2978–2983, London, UK, September 2013.

[4] Sandra Herker, Wolfgang Kiess, Xueli An, and Andreas Kirstädter. On the trade-off between cost and availability of virtual networks. In *2014 IFIP Networking Conference*, pages 1–9, Trondheim, June 2014.

[5] Sandra Herker, Xueli An, Wolfgang Kiess, and Andreas Kirstädter. Evaluation of Data-Center Architectures for Virtualized Network Functions. In *IEEE International Conference on Communications Workshop on Cloud Computing Systems, Networks, and Applications (ICC Workshop)*, London, UK, June 2015.

[6] Sandra Herker, Xueli An, Wolfgang Kiess, Sergio Beker, and Andreas Kirstädter. Data-Center Architecture Impacts on Virtualized Network Functions Service Chain Embedding with High Availability Requirements. In *IEEE Global Communications Conference Workshops (GLOBECOM)*, San Diego, December 2015.

[7] Xueli An-de Luca, Ashiq Khan, and Sandra Herker. Method for mapping a network topology request to a physical network, computer program product, mobile communication system, and network configuration platform, November 2014. URI: `http://www.freepatentsonline.com/EP2804343A1.html`.

[8] Xueli An-de Luca, Ashiq Khan, and Sandra Herker. Method for mapping network topology request to physical network, computer program product, mobile communication system and network configuration platform, December 2014. URI: `http://www.freepatentsonline.com/JP2014225872.html`.

[9] A. Nakao. Network Virtualization as Foundation for Enabling New Network Architectures and Applications. *IEICE Transactions on Communications*, 93:454–457, 2010.

[10] 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.

[11] N. M. Mosharaf Kabir Chowdhury and Raouf Boutaba. Network virtualization: state of the art and research challenges. *Communications Magazine, IEEE*, 47:20–26, July 2009.

[12] N.M. Mosharaf Kabir Chowdhury and Raouf Boutaba. A survey of network virtualization. *Computer Networks*, 54:862–876, April 2010.

[13] Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt, and Andrew Warfield. Xen and the Art of Virtualization. In *Proceedings of the Nineteenth ACM Symposium on Operating Systems Principles*, SOSP '03, New York, NY, USA, 2003. ACM.

[14] Nick McKeown et al. OpenFlow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38:69–74, March 2008.

[15] 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. ACM, 2009.

[16] Ashiq Khan, Dan Jurca, and Wolfgang Kellerer. The Reconfigurable Mobile Network. In *IEEE ICC Workshop on Advances in Mobile Networking (ICC Workshop)*, Kyoto, Japan, June 2011.

[17] M Chiosi et al. Network functions virtualisation: An introduction, benefits, enablers, challenges and call for action. Technical report, White paper ETSI, 2012.

[18] Hassan Hawilo, Abdallah Shami, Maysam Mirahmadi, and Rasool Asal. NFV: state of the art, challenges, and implementation in next generation mobile networks (vEPC). *IEEE Network*, 28(6):18–26, November 2014.

[19] Alcatel-Lucent. Network Functions Virtualization - Challenges and Solutions. Technical report, Strategic White Paper, Alcatel-Lucent, 2013.

[20] ETSI. Network Functions Virtualisation (NFV); Resiliency Requirements. Technical report, ETSI, 2014.

[21] Wind River Systems. NFV: The myth of application-level high availability. Technical report, White Paper, 2015.

[22] ETSI. Network Functions Virtualization (NFV); Architectural Framework. Technical report, ETSI, 2013.

[23] ETSI. Network Functions Virtualization (NFV); Use Cases. Technical report, ETSI, 2013.

[24] Martin L. Shooman. *Reliability of Computer Systems and Networks: Fault Tolerance, Analysis, and Design*. John Wiley & Sons, 2002.

[25] James PG Sterbenz, David Hutchison, Egemen K Çetinkaya, Abdul Jabbar, Justin P Rohrer, Marcus Schöller, and Paul Smith. Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines. *Computer Networks*, 54(8):1245–1265, 2010.

[26] Eric Bauer and Randee Adams. *Reliability and availability of cloud computing*. John Wiley & Sons, 2012.

[27] Susan Stanley. MTBF, MTTR, MTTF & FIT Explanation of Terms. *IMC Network*, pages p1–6, 2011.

[28] M. Krasich. How to estimate and use MTTF/MTBF would the real MTBF please stand up? In *Reliability and Maintainability Symposium, 2009. RAMS 2009. Annual*, pages 353–359, January 2009.

[29] S. Verbrugge, D. Colle, P. Demeester, R. Huelsermann, and M. Jaeger. General availability model for multilayer transport networks. In *International Workshop on Design of Reliable Communication Networks (DRCN)*, 2005.

[30] Yoichi Maeda, Francesco Montalti, et al. Optical fibres, cables and systems. Technical report, Technical report, ITU, 2009. URI http://www.itu.int/dms_pub/itu-t/opb/hdb/T-HDB-OUT.10-2009-1-PDF-E.pdf, 2009.

[31] Jun Zheng and Hussein T Mouftah. *Optical WDM networks: concepts and design principles*. John Wiley & Sons, 2004.

[32] Jing Zhang and B Mukheriee. A review of fault management in wdm mesh networks: basic concepts and research challenges. *IEEE Network*, 18(2):41–48, 2004.

[33] Mohammad Al-Fares, Alexander Loukissas, and Amin Vahdat. A scalable, commodity data center network architecture. *ACM SIGCOMM Computer Communication Review*, 38(4):63–74, 2008.

[34] Kashif Bilal, Saif Ur Rehman Malik, Osman Khalid, Abdul Hameed, Enrique Alvarez, Vidura Wijaysekara, Rizwana Irfan, Sarjan Shrestha, Debjyoti Dwivedy, Mazhar Ali, et al. A taxonomy and survey on Green Data Center Networks. *Future Generation Computer Systems*, 36:189–208, 2014.

[35] Kashif Bilal, Samee U Khan, Limin Zhang, Hongxiang Li, Khizar Hayat, Sajjad A Madani, Nasro Min-Allah, Lizhe Wang, Dan Chen, Majid Iqbal, et al. Quantitative comparisons of the state-of-the-art data center architectures. *Concurrency and Computation: Practice and Experience*, 25(12):1771–1783, 2013.

[36] M Faizul Bari, Raouf Boutaba, Rafael Esteves, Lisandro Z Granville, Maxim Podlesny, Md Golam Rabbani, Qi Zhang, and Mohamed Faten Zhani. Data center network virtualization: A survey. *Communications Surveys & Tutorials, IEEE*, 15(2):909–928, 2013.

[37] Chuanxiong Guo, Guohan Lu, Dan Li, Haitao Wu, Xuan Zhang, Yunfeng Shi, Chen Tian, Yongguang Zhang, and Songwu Lu. BCube: a high performance, server-centric network architecture for modular data centers. *ACM SIGCOMM Computer Communication Review*, 39(4):63–74, 2009.

[38] Chuanxiong Guo, Haitao Wu, Kun Tan, Lei Shi, Yongguang Zhang, and Songwu Lu. DCell: a scalable and fault-tolerant network structure for data centers. *ACM SIGCOMM Computer Communication Review*, 38(4):75–86, 2008.

[39] Athina Markopoulou, Gianluca Iannaccone, Supratik Bhattacharyya, Chen-Nee Chuah, Yashar Ganjali, and Christophe Diot. Characterization of Failures in an Operational IP Backbone Network. *IEEE/ACM Transactions on Networking*, 16(4):749–762, 2008.

[40] 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. ACM, 2002.

[41] Craig Labovitz, Abha Ahuja, and Farnam Jahanian. Experimental study of internet stability and backbone failures. In *Twenty-Ninth Annual International Symposium on Fault-Tolerant Computing*, pages 278–285. IEEE, 1999.

[42] Daniel Turner, Kirill Levchenko, Alex C Snoeren, and Stefan Savage. California fault lines: understanding the causes and impact of network failures. *ACM SIGCOMM Computer Communication Review*, 41(4):315–326, 2011.

[43] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, Chen-Nee Chuah, and C. Diot. Characterization of failures in an IP backbone. In *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 4, pages 2307–2317, March 2004.

[44] Dan Crawford. Fiber Optics Cable Dig-ups, Causes and Cures. *Proc. Network Reliability: A Report to the Nation - Compendium of Technical Papers, National Engineering Consortium*, June 1993.

[45] ALCOA FUJIKURA LTD. Reliability of fiber optic cable systems: Buried fiber optic cable, optical groundwire cable, all dielectric, self supporting cable. Technical report, May 2001. URI: `http://www.southern-telecom.com/solutions/afl-reliability.pdf`.

[46] I.B. Heard. Availability and cost estimation of secured FTTH architectures. In *International Conference on Optical Network Design and Modeling (ONDM)*, 2008.

[47] Ralf Huelsermann, Monika Jaeger, Arie M. C. A. Koster, Sebastian Orlowski, Roland Wessaely, and Adrian Zymolka. Availability and Cost Based Evaluation of Demand-wise Shared Protection. In *ITG Symposium on Photonic Networks*, 2006.

[48] D. A A Mello, D.A. Schupke, M. Scheffel, and H. Waldman. Availability maps for connections in WDM optical networks. In *International Workshop on Design of Reliable Communication Networks (DRCN)*, 2005.

[49] D. A. Schupke, A. Autenrieth, and T. Fischer. Survivability of Multiple Fiber Duct Failures. In *International Workshop on the Design of Reliable Communication Networks (DRCN)*, 2001.

[50] M. Tornatore, G. Maier, and A. Pattavina. Availability Design of Optical Transport Networks. *IEEE Journal on Selected Areas in Communications*, 23(8):1520–1532, 2005.

[51] Rahul Potharaju and Navendu Jain. When the Network Crumbles: An Empirical Study of Cloud Network Failures and Their Impact on Services. In *Proceedings of the 4th Annual Symposium on Cloud Computing*, SOCC '13, New York, NY, USA, 2013. ACM.

[52] Phillipa Gill, Navendu Jain, and Nachiappan Nagappan. Understanding Network Failures in Data Centers: Measurement, Analysis, and Implications. In *Proceedings of the ACM SIGCOMM Conference*, SIGCOMM '11, pages 350–361. ACM, 2011.

[53] Kashi Venkatesh Vishwanath and Nachiappan Nagappan. Characterizing Cloud Computing Hardware Reliability. In *Proceedings of the 1st ACM Symposium on Cloud Computing*, SoCC '10, pages 193–204, New York, NY, USA, 2010. ACM.

[54] Bianca Schroeder and Garth A. Gibson. Disk Failures in the Real World: What Does an MTTF of 1,000,000 Hours Mean to You? In *Proceedings of the USENIX Conference on File and Storage Technologies*, FAST '07, Berkeley, CA, USA, 2007. USENIX Association.

[55] Wei Deng, Hai Jin, Xiaofei Liao, Fangming Liu, Li Chen, and Haikun Liu. Lifetime or Energy: Consolidating Servers with Reliability Control in Virtualized Cloud Datacenters. In *Proceedings of the 2012 IEEE 4th International Conference on Cloud Computing Technology and Science (CloudCom)*, CLOUDCOM '12, pages 18–25, Washington, DC, USA, 2012. IEEE.

[56] Robert Birke, Ioana Giurgiu, Lydia Y. Chen, Dorothea Wiesmann, and Ton Engbersen. Failure Analysis of Virtual and Physical Machines: Patterns, Causes and Characteristics. In *Proceedings of the 2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, DSN '14, pages 1–12, Washington, DC, USA, 2014. IEEE.

[57] Cisco systems, December 2015. URI: `http://www.cisco.com/`.

[58] Intel server systems, December 2015. URI: `http://www.intel.com/content/www/us/en/server-systems/intel-server-systems.html`.

[59] Andreas Fischer, Juan F. Botero, Michael Duelli, Daniel Schlosser, Xavier Hesselbach, and Hermann De Meer. ALEVIN - A Framework to Develop, Compare, and Analyze Virtual Network Embedding Algorithms. *Electronic Communications of the EASST*, 37:1–12, 2011.

[60] Anath Fischer, Juan Felipe Botero, Michael Till Beck, Hermann De Meer, and Xavier Hesselbach. Virtual network embedding: A survey. *Communications Surveys & Tutorials, IEEE*, 15(4):1888–1906, 2013.

[61] Flavio Esposito, Ibrahim Matta, and Vatche Ishakian. Slice embedding solutions for distributed service architectures. *ACM Computing Surveys*, 46(1), 2013.

[62] Minlan Yu, Yung Yi, Jennifer Rexford, and Mung Chiang. Rethinking virtual network embedding: substrate support for path splitting and migration. *ACM SIGCOMM Computer Communication Review*, 38:17–29, March 2008.

[63] David G. Andersen. Theoretical Approaches to Node Assignment. Unpublished Manuscript, December 2002.

[64] Ines Houidi, Wajdi Louati, Walid Ben Ameur, and Djamal Zeghlache. Virtual network provisioning across multiple substrate networks. *Computer Networks*, 55:1011–1023, March 2011.

[65] Jens Lischka and Holger Karl. A Virtual Network Mapping Algorithm Based on Subgraph Isomorphism Detection. In *Proceedings of the 1st ACM Workshop on Virtualized Infrastructure Systems and Architectures*, VISA '09, pages 81–88. ACM, 2009.

[66] Robert Ricci, Chris Alfeld, and Jay Lepreau. A solver for the network testbed mapping problem. *ACM SIGCOMM Computer Communication Review*, 33:65–81, April 2003.

[67] N.M.M.K. Chowdhury, M.R. Rahman, and R. Boutaba. Virtual Network Embedding with Coordinated Node and Link Mapping. In *Proceedings IEEE INFOCOM*, pages 783–791, April 2009.

[68] Y. Zhu and M. Ammar. Algorithms for Assigning Substrate Network Resources to Virtual Network Components. In *Proceedings IEEE INFOCOM*, pages 1–12, April 2006.

[69] W. Szeto, Y. Iraqi, and R. Boutaba. A multi-commodity flow based approach to virtual network resource allocation. In *IEEE Global Communications Conference (GLOBECOM)*, volume 6, pages 3004–3008, December 2003.

[70] David Eppstein. Finding the k shortest paths. In *Proceedings of 35th Annual Symposium on Foundations of Computer Science*, pages 154–165. IEEE, 1994.

[71] S. Ramamurthy, L. Sahasrabuddhe, and B. Mukherjee. Survivable WDM mesh networks. *Journal of Lightwave Technology*, 21(4):870 – 883, April 2003.

[72] Fernando A Kuipers. An overview of algorithms for network survivability. *ISRN Communications and Networking*, 2012.

[73] Jing Lu and Jonathan Turner. Efficient Mapping of Virtual Networks onto a Shared Substrate. Technical report, Washington University in St. Louis, 2006. URI: `http://www.arl.wustl.edu/~{}jl1/research/tech_report_2006.pdf`.

[74] Hyungjin Kim and Sanghwan Lee. Greedy virtual network embedding under an exponential cost function. In *International Conference on Information Networking (ICOIN)*, February 2012.

[75] Mosharaf Chowdhury, Fady Samuel, and Raouf Boutaba. PolyViNE: Policy-based Virtual Network Embedding Across Multiple Domains. In *Proceedings of the Second ACM SIGCOMM Workshop on Virtualized Infrastructure Systems and Architectures*, VISA '10, pages 49–56, New York, NY, USA, 2010. ACM.

[76] Bo Lv, Zhenkai Wang, Tao Huang, Jianya Chen, and Yunjie Liu. Virtual Resource Organization and Virtual Network Embedding across Multiple Domains. In *Proceedings of the International Conference on Multimedia Information Networking and Security*, MINES '10, pages 725–728, Washington, DC, USA, 2010. IEEE Computer Society.

[77] Christoph Werle, Panagiotis Papadimitriou, Ines Houidi, Wajdi Louati, Djamal Zeghlache, Roland Bless, and Laurent Mathy. Building Virtual Networks Across Multiple Domains. In *Proceedings of the ACM SIGCOMM Conference*, SIGCOMM '11, pages 412–413, New York, NY, USA, 2011. ACM.

[78] Yufeng Xin, Ilia Baldine, Anirban Mandal, Chris Heermann, Jeff Chase, and Aydan Yumerefendi. Embedding Virtual Topologies in Networked Clouds. In *Proceedings of the 6th International Conference on Future Internet Technologies*, CFI '11, pages 26–29, New York, NY, USA, 2011. ACM.

[79] Juan Segovia, Eusebi Calle, Pere Vila, Jose Marzo, and Janos Tapolcai. Topology-focused availability analysis of basic protection schemes in optical transport networks. *Journal of Optical Networking*, 7(4), 2008.

[80] Yu Liu, D. Tipper, and P. Siripongwutikorn. Approximating optimal spare capacity allocation by successive survivable routing. *IEEE/ACM Transactions on Networking*, 13(1):198 – 211, February 2005.

[81] Muntasir Raihan Rahman, Issam Aib, and Raouf Boutaba. Survivable virtual network embedding. In *Proceedings of the 9th IFIP TC 6 international conference on Networking*, NETWORKING'10, pages 40–52, 2010.

[82] Tao Guo, Ning Wang, K. Moessner, and R. Tafazolli. Shared Backup Network Provision for Virtual Network Embedding. In *IEEE International Conference on Communications (ICC)*, pages 1–5, June 2011.

[83] Yang Chen, Jianxin Li, Tianyu Wo, Chunming Hu, and Wantao Liu. Resilient virtual network service provision in network virtualization environments. In *2010 IEEE 16th International Conference on Parallel and Distributed Systems (ICPADS)*, pages 51–58. IEEE, 2010.

[84] Hongfang Yu, V. Anand, Chunming Qiao, and Hao Di. Migration based protection for virtual infrastructure survivability for link failure. In *Conference on Optical Fiber Communication/National Fiber Optic Engineers Conference (OFC/NFOEC)*, pages 1–3, March 2011.

[85] J. Shamsi and M. Brockmeyer. QoSMap: Achieving Quality and Resilience through Overlay Construction. In *4th International Conference on Internet and Web Applications and Services, ICIW '09*, pages 58 –67, May 2009.

[86] Xian Zhang, C. Phillips, and Xiuzhong Chen. An overlay mapping model for achieving enhanced QoS and resilience performance. In *3rd International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT 2011)*, pages 1–7, October 2011.

[87] Xian Zhang and Chris Phillips. A novel heuristic for overlay mapping with enhanced resilience and QoS. In *IET International Conference on Communication Technology and Application (ICCTA)*, pages 540–545, October 2011.

[88] Rodrigo R Oliveira, Daniel S Marcon, Leonardo Richter Bays, Miguel C Neves, Luciana Salete Buriol, Luciano Paschoal Gaspary, and Marinho Pilla Barcellos. No more backups: Toward efficient embedding of survivable virtual networks. In *IEEE International Conference on Communications (ICC)*, pages 2128–2132. IEEE, 2013.

[89] Hongfang Yu, V. Anand, Chunming Qiao, and Gang Sun. Cost Efficient Design of Survivable Virtual Infrastructure to Recover from Facility Node Failures. In *IEEE International Conference on Communications (ICC)*, pages 1–6, June 2011.

[90] Chunming Qiao, Bingli Guo, Shanguo Huang, Jianping Wang, Ting Wang, and Wanyi Gu. A novel two-step approach to surviving facility failures. In *Conference on Optical Fiber Communication/National Fiber Optic Engineers Conference (OFC/NFOEC)*, pages 1–3, March 2011.

[91] Qian Hu, Yang Wang, and Xiaojun Cao. Location-constrained survivable network virtualization. In *IEEE Sarnoff Symposium (SARNOFF)*, pages 1 –5, May 2012.

[92] Hongfang Yu, Chunming Qiao, V. Anand, Xin Liu, Hao Di, and Gang Sun. Survivable Virtual Infrastructure Mapping in a Federated Computing and Networking System under Single Regional Failures. In *IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, December 2010.

[93] Wai-Leong Yeow, Cédric Westphal, and Ulaş Kozat. Designing and embedding reliable virtual infrastructures. In *Proceedings of the Second ACM SIGCOMM Workshop on Virtualized Infrastructure Systems and Architectures*, VISA '10, pages 33–40. ACM, 2010.

[94] 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. Springer-Verlag, 2012.

[95] Abdallah Jarray, Yihong Song, and Ahmed Karmouch. p-Cycle-based node failure protection for survivable virtual network embedding. In *2013 IFIP Networking Conference*, pages 1–9. IEEE, 2013.

[96] M Pourvali, H Bai, F Gu, K Shaban, M Naeini, J Crichigno, M Hayat, Sharifullah Khan, and N Ghani. Virtual network mapping for cloud services under probabilistic regional failures. In *IEEE 3rd International Conference on Cloud Networking (CloudNet)*, pages 407–412. IEEE, 2014.

[97] S Baucke and C Görg. D-3.1. 1 Virtualisation Approach: Concept. Technical report, FP7 4WARD Project, 2009.

[98] Wenzhu Zou, Milena Janic, Robert Kooij, and Fernando Kuipers. On the availability of networks. *BroadBand Europe*, pages 3–6, 2007.

[99] G. Semaan. Designing Networks with the Optimal Availability. In *Conference on Optical Fiber Communication/National Fiber Optic Engineers Conference (OFC/NFOEC)*, pages 1–6, Feb 2008.

[100] M. Clouqueur and W.D. Grover. Availability analysis of span-restorable mesh networks. *IEEE Journal on Selected Areas in Communications*, 20(4):810–821, 2002.

[101] Song Yang, S. Trajanovski, and F.A. Kuipers. Availability-based path selection. In *2014 6th International Workshop on Reliable Networks Design and Modeling (RNDM)*, pages 39–46, Nov 2014.

[102] Rafael L Gomes, Luiz F Bittencourt, and Edmundo RM Madeira. A Bandwidth-Feasibility Algorithm for Reliable Virtual Network Allocation. In *IEEE 28th International Conference on Advanced Information Networking and Applications (AINA)*, pages 504–511. IEEE, 2014.

[103] Yiheng Chen, Sara Ayoubi, and Chadi Assi. CORNER: COst-Efficient and Reliability-Aware Virtual NEtwork Redesign and Embedding. In *IEEE 3rd International Conference on Cloud Networking (CloudNet)*, pages 356–361. IEEE, 2014.

[104] Riccardo Guerzoni, Zoran Despotovic, Riccardo Trivisonno, and Ishan Vaishnavi. Modeling Reliability Requirements in Coordinated Node and Link Mapping. In *IEEE 33rd International Symposium on Reliable Distributed Systems (SRDS)*, pages 321–330. IEEE, 2014.

[105] W.D. Grover and A. Sack. High availability survivable networks: When is reducing MTTR better than adding protection capacity? In *International Workshop on Design and Reliable Communication Networks (DRCN)*, 2007.

[106] Tracey Cohen and Russell Southwood. Extending open access to national fibre backbones in developing countries. In *8th Global Symposium for Regulators ITU*, 2008.

[107] Greg Taylor, Paul Ray, and Memo. LWA Data Communications - The Fiber Option. Technical report, LWA, 2008.

[108] John Ellershaw, Jennifer Riding, Alan Lee, An Vu Tran, Lin Jie Guan, Rod Tucker, Timothy Smith, and Erich Stumpf. Deployment costs of rural broadband technologies. *Telecommunications Journal of Australia*, 59(2), 2009.

[109] George McGuire. Enhanced cost solutions for buried fiber installation. Technical report, 2009.

[110] J. W. Suurballe. Disjoint paths in a network. *Networks*, 4(2):125–145, 1974.

[111] A. Makhorin. GLPK (GNU Linear Programming Kit). Available at `http://www.gnu.org/software/glpk/glpk.html`.

[112] S. Knight, H.X. Nguyen, N. Falkner, R. Bowden, and M. Roughan. The Internet Topology Zoo. *IEEE Journal on Selected Areas in Communications*, 29(9), 2011. URI: `http://www.topology-zoo.org/`.

[113] SCOPE Alliance. Telecom Grade cloud Computing v1.0. Technical report, White paper, 2011. URI: http://scope-alliance.org/sites/default/files/documents/cloudComputing_Scope_1.0.pdf.

[114] Xiaoqiao Meng, Vasileios Pappas, and Li Zhang. Improving the scalability of data center networks with traffic-aware virtual machine placement. In *Proceedings IEEE INFOCOM*, pages 1–9. IEEE, 2010.

[115] Yueping Zhang, Ao-Jan Su, and Guofei Jiang. Evaluating the impact of data center network architectures on application performance in virtualized environments. In *2010 18th International Workshop on Quality of Service (IWQoS)*, pages 1–5. IEEE, 2010.

[116] Joe Wenjie Jiang, Tian Lan, Sangtae Ha, Minghua Chen, and Mung Chiang. Joint VM placement and routing for data center traffic engineering. In *Proceedings IEEE INFOCOM*, pages 2876–2880. IEEE, 2012.

[117] Xin Li, Jie Wu, Shaojie Tang, and Sanglu Lu. Let's stay together: Towards traffic aware virtual machine placement in data centers. In *Proceedings IEEE INFOCOM*, pages 1842–1850. IEEE, 2014.

[118] Jielong Xu, Jian Tang, Kevin Kwiat, Weiyi Zhang, and Guoliang Xue. Survivable Virtual Infrastructure Mapping in Virtualized Data Centers. In *2012 IEEE 5th International Conference on Cloud Computing (CLOUD)*, pages 196 –203. IEEE, June 2012.

[119] Fumio Machida, Masahiro Kawato, and Yoshiharu Maeno. Redundant virtual machine placement for fault-tolerant consolidated server clusters. In *IEEE Network Operations and Management Symposium (NOMS)*, pages 32–39. IEEE, 2010.

[120] Md Golam RABBANI, Mohamed Faten ZHANI, and Raouf BOUTABA. On Achieving High Survivability in Virtualized Data Centers. *IEICE TRANSACTIONS on Communications*, 97(1):10–18, 2014.

[121] Qi Zhang, M.F. Zhani, M. Jabri, and R. Boutaba. Venice: Reliable Virtual Data Center Embedding in Clouds. In *Proceedings IEEE INFOCOM*, pages 289–297, April 2014.

[122] Wenting Wang, Haopeng Chen, and Xi Chen. An availability-aware virtual machine placement approach for dynamic scaling of cloud applications. In *9th International Conference on Ubiquitous Intelligence & Computing and 9th International Conference on Autonomic & Trusted Computing (UIC/ATC)*, pages 509–516. IEEE, 2012.

[123] Deepal Jayasinghe, Calton Pu, Tamar Eilam, Malgorzata Steinder, Ian Whally, and Ed Snible. Improving performance and availability of services hosted on IaaS clouds with structural constraint-aware virtual machine placement. In *IEEE International Conference on Services Computing (SCC)*, pages 72–79. IEEE, 2011.

[124] Marcus Scholler, Martin Stiemerling, Andreas Ripke, and Roland Bless. Resilient deployment of virtual network functions. In *5th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, pages 208–214. IEEE, 2013.

[125] Arista networks, December 2015. URI: http://www.arista.com/.

[126] Dell poweredge servers, December 2015. URI: `http://www.dell.com/us/business/p/poweredge-blade-servers`.

[127] Mellanox adapters, December 2015. URI: `http://www.mellanoxstore.com/categories/adapters.html`.

[128] Lucian Popa, Sylvia Ratnasamy, Gianluca Iannaccone, Arvind Krishnamurthy, and Ion Stoica. A Cost Comparison of Datacenter Network Architectures. In *Proceedings of the 6th International COnference*, Co-NEXT '10. ACM, 2010.

[129] A. Gulati, A. Holler, M. Ji, G. Shanmuganathan, C. Waldspurger, and X. Zhu. VMware Distributed Resource Management: Design, Implementation, and Lessons Learned. In *VMware Technical Journal*, Spring 2012.

[130] Arjun Singh, Joon Ong, Amit Agarwal, Glen Anderson, Ashby Armistead, Roy Bannon, Seb Boving, Gaurav Desai, Bob Felderman, Paulie Germano, et al. Jupiter Rising: A Decade of Clos Topologies and Centralized Control in Google's Datacenter Network. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, SIGCOMM '15, pages 183–197. ACM, 2015.

[131] Lucian Popa, Norbert Egi, Sylvia Ratnasamy, and Ion Stoica. Building Extensible Networks with Rule-based Forwarding. In *Proceedings of the USENIX Conference on Operating Systems Design and Implementation*, OSDI'10. USENIX Association, 2010.

[132] D5.3: Application Development and Deployment. Technical report, FP7 CHANGE Project, 2013.

[133] Joao Martins, Mohamed Ahmed, Costin Raiciu, Vladimir Olteanu, Michio Honda, Roberto Bifulco, and Felipe Huici. ClickOS and the Art of Network Function Virtualization. In *Proceedings of the USENIX Conference on Networked Systems Design and Implementation*, NSDI'14. USENIX Association, 2014.

[134] Haitao Wu, Guohan Lu, Dan Li, Chuanxiong Guo, and Yongguang Zhang. MDCube: A High Performance Network Structure for Modular Data Center Interconnection. In *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies*, CoNEXT '09, pages 25–36, New York, NY, USA, 2009. ACM.

[135] Dong Lin, Yang Liu, Mounir Hamdi, and Jogesh Muppala. Hyper-BCube: A scalable data center network. In *IEEE International Conference on Communications (ICC)*, pages 2918–2923. IEEE, June 2012.

[136] Dan Li, Chuanxiong Guo, Haitao Wu, Kun Tan, Yongguang Zhang, and Songwu Lu. FiConn: Using Backup Port for Server Interconnection in Data Centers. In *Proceedings IEEE INFOCOM*, pages 2276–2285. IEEE, April 2009.

Acknowledgments

First of all I would like to express my gratitude to DOCOMO Euro-Labs for giving me the chance to complete my thesis and perform research on various interesting and challenging subjects. First and foremost I would like to thank Prof. Andreas Kirstädter for his supervision throughout my work. I appreciate all his time, aspiring guidance and invaluably constructive criticism. Furthermore, I would like to thank Prof. Thomas Bauschert for being my second examiner.

Special thanks go to my DOCOMO supervisor Xueli An for her valuable help, advice, and comments on different aspects of my thesis. She gave me her trust, was very supportive through my thesis and was always available for discussions and idea exchange. Further, I thank Wolfgang Kiess, Ashiq Khan and Ishan Vaishnavi for their great and invaluable support during my thesis. I had many useful discussions and entertaining moments with my office-mates Bo Fu, Marwa El Hefnawy, Emmanuel Ternon and Thorsten Biermann, I thank all of them. Also, I would like to thank all others from Networking Group and all other colleagues and friends from DOCOMO Euro-Labs. They made my time at DOCOMO Euro-Labs a valuable and unforgettable experience.

Finally, my deepest gratitude goes to my parents for their unconditional support, care, and love in addition to their trust in me. I am lucky to have them and I dedicate this work to them. Special thanks to my family and friends. They were always there for me and they always supported me in every decision I took.