Optimal resource allocation in Virtual Network Embedding

M Hollestein

orcid.org/0000-0001-5918-6864

Dissertation submitted in fulfilment of the requirements for the degree *Master of Engineering in Computer and Electronic Engineering* at the North-West University

Supervisor: Prof SE Terblanche

Co-supervisor: Prof MJ Grobler

Graduation ceremony: May 2019

Student number: 24215392

Optimal Resource Allocation in Virtual Network Embedding

Dissertation submitted in fulfilment of the requirements for the degree
Master of Engineering **in Computer Engineering at the Potchefstroom campus of the North-West University**

M. Hollestein

24215392

Supervisor: Prof. S.E. Terblanche
Supervisor: Prof. M.J. Grobler

November 2018

Declaration

I, Marja Hollestein hereby declare that the dissertation entitled "Optimal Resource Allocation in Virtual Network Embedding" is my own original work and has not already been submitted to any other university or institution for examination.

M. Hollestein

Student number: 24215392

Signed on the 20th day of November 2018 at Potchefstroom.

Acknowledgements

To commence with, I must thank GOD, the Almighty, for providing me with good health, courage, inspiration and zeal to complete this study.

I want to thank my thesis advisers Prof. MJ Grobler and Prof. SE Terblanche for the valuable comments, remarks and engagement through the learning process of this master thesis and for introducing me to this topic.

I am indebted to the Telkom Centre of Excellence for providing me with the platform to conduct my research and for merSETA for providing me with the financial assistance to enable me to complete this dissertation.

I want to thank the TeleNet research group for the friendship and support through this process.

I must express my very profound gratitude to my parents for allowing me the opportunity to follow my dreams wherever they go.

Finally, I am forever appreciative to Gerhard van Zyl for the unfailing support and continuous encouragement throughout the process of researching and writing this thesis; I will always treasure your motivation.

This accomplishment would not have been possible without you all. Thank you.

Abstract

Optimal resource allocation has become a focus in the technological environment due to the high costs associated with physical infrastructure and space. Furthermore, Network Operators (NO)s have to keep up with the rate of change in this digital age. This led to the creation of network virtualization. Virtualizing a network means that the functionality of the network becomes independent of the physical hardware that supports it. A significant advantage is that the same physical server can be used for several different purposes - depending on the software installed - to enable faster and more efficient communication.

To prevent stagnation of internet infrastructure, Virtual Network Embedding (VNE) has emerged as a promising component of the future Internet. Virtual networks (VNs) still requires a substrate infrastructure and over-provisioning and non-optimal resource allocation are, therefore, still a critical consideration when implementing VNEs.

Virtual network embedding refers to the instance where multiple virtual networks are hosted on the same substrate network (SN). VNE is thus the instance where various network nodes are installed on the same infrastructure; this enables shared capacity between them. When a change in demand is observed it is not necessary to purchase new network hardware, it is merely a matter of changing the software to balance the capacity between nodes dynamically.

Internet Service Providers (ISPs) are currently viewed as two separate entities in order to provide dedicated networks and services separately. The Infrastructure Provider (InP)and the Service Provider (SP) are the central division in this new business model and rely on modern structures to provide services to clients.

In this paper, a Mixed Integer Linear Program is proposed to model a VNE problem which includes off-line stochastic resource allocation (SRA). Two case studies are used in this dissertation. Case study A is used to verify the model and is a simple experiment with only ten nodes and three requests. Each of these requests comprises of two scenarios. The case study is completed for a worst-case version, as well as a stochastic version of the problem. This condensed version is used to ensure the calculations - to verify optimality - are possible by hand.

Case study B is a more extensive case study with twenty nodes in the substrate infrastructure. There are fourteen requests with three scenarios each. This case study is used to validate the proposed model and is also performed in the worst case as well as the stochastic version. The reason a worst-case version is performed for each case study is to illustrate the improvement SRA can provide since the worst-case version of the case study is the manner in which many older works complete their VNE.

Using the increase in model size the scalability is also tested to some extent, but cannot be proven with such little data. The results conclude that a Mixed Integer Linear Program can be successfully used to implement a stochastic embedding approach considering VNE.

It is evident from the results of the study that this approach has the advantage of increased resource allocation which is found to be financially beneficial for a supplier of the service. This highlights the gains that a service provider can obtain, that is preferable in a financial sense, as well as a positive impact on the environment by using less physical resources.

Keywords: *VNE, SRA, Optimisation, Resource Allocation*

Contents

List of Figures

List of Tables

List of Acronyms

VNE Virtual Network Embedding

VN Virtual Network

VNoM Virtual Node Mapping

VLiM Virtual Link Mapping

SRA Stochastic Resource Allocation

NFV Network Function Virtualization

SDN Software Defined Networking

CapEx Capital Expenditure

OpEx Operational Expenditure

NO Network Operator

SN Substrate Network

VN Virtual Network

VNR Virtual Network Request

QoS Quality of Service

QoNE Quality of Network Economics

QoR Quality of Resilience

ISP Internet Service Provider

InP Infrastructure Provider

SP Service Provider

VNO Virtual Network Operator

VNP Virtual Network Provider

COTS commercial off-the-shelf

IaaS Infrastructure as a Service

RO Robust Optimisation

SVNE Survivable Virtual Network Embedding

EEVNE Energy Efficient Virtual Network Embedding

VM Virtual Machine

LP Linear Programming

RAP Resource Allocation Problem

VON Virtual Optical Network

MCF Multi-commodity Flow Problem

MIP Mixed Integer Programming

BFS breadth-first search

ILP Integer Linear Programming

MILP Mixed Integer Linear Programming

NP Non-Deterministic Polynomial-Time

IP Integer Programming

Chapter 1

Introduction

This chapter is the introductory chapter of the dissertation where the scope of the dissertation was explored as well as the importance of the research conducted. Some essential context is also provided.

1.1 Background

A network is a system of nodes and links that are interconnected to form a more extensive system. Networks have a strong dependency on the hardware (physical equipment) needed to form the system, which means that a network is a costly and rigid infrastructure to operate. Implementing changes to the network can be difficult as it can be time-consuming and in order to include any new functionality, time and money are needed. All these factors cause the flexibility of a traditional network to be limited [1].

Optimal resource allocation has become a focus in the technological environment due to the cost associated with physical space. Furthermore, network operators also need to be adaptable in order to keep up with the rate of change in the digital age [2]. This has led to the creation of network virtualization.

Virtualizing a network means that the software is separated from the networking equipment. Thus the functionality of the network become more independent of the physical hardware that supports it. A significant advantage is that the same physical server can be used for several different purposes depending on the software installed to enable faster and more efficient communication.

Han et al. [3] argues that Network Function Virtualization (NFV) enables three major deviations when compared to the archaic manner of performing network functionality. These differences are given as:

- Separation of software from hardware, which allows the software to evolve and be developed independently of the hardware as well as allow the hardware to evolve autonomously from the software.
- Flexible deployment of network functions, since the use of NFV allows the automatic deployment of network functionalities in the form of software on a consortium of hardware resources and likewise, these hardware resources can execute different functions at different times as well as in different data centres.
- Dynamic service provisioning, the NFV performance can be scaled on a dynamic basis, or even an increase as demand increases basis with ragged control based on the current network conditions.

The authors also state that NFV interrelates to other emerging technologies, with specific reference to Software Defined Networking (SDN). When the control plane is dissociated from the underlying data plane, and the control functions are consolidated into a logically centralised controller, it is referred to as SDN. These two solutions can be combined to increase the aggregated value. NFV and SDN can be used in a symbiotic relationship since they are seen as mutually beneficial. This combination of solutions is considered highly complementary to each other and share some features, such as the promotion of innovation, creativity, openness and competitiveness. The authors stress that virtualization and deployment of network functions do not rely on SDN technologies, as is true in the reversed case. An example of the combination of these two technologies that the authors provide is that SDN can support NFV to enhance its performance, facilitate its operation, and simplify the compatibility with legacy deployments [3].

Hardware resources in virtual networks must be allocated with the utmost care to ensure that a Virtual Network (VN) can operate at the same speed as traditional rigid networks, as well as to achieve predictable and high levels of performance [4]. A virtual network is a mouldable structure where functions can be deployed on demand quickly and efficiently [5]; thus from a financial perspective, virtual networks will be more efficient than traditional ones since the less physical hardware is needed to house the same network functions and expansion is less time consuming [1].

It is also possible for multiple virtual networks to be hosted on the same substrate network at the same time [1]. This multi-tenancy is done through resource allocation. Each time a user needs to use their virtual network a request is made of the virtual embedded network system. This request is received, and specific physical resources are allocated to this specific network request. The system allocates certain nodes and links to this specific request in order to accomplish the usage goal that this virtual network request has.

In certain instances, the allocation of resources in virtual networks is an inefficient and redundant process which causes the same substrate resources to host less virtual networks than possible. If the allocation of virtual resources (which are hosted on a substrate network) can be optimised, it may be possible to optimally use the same physical resources to host more virtual networks than when just allocating resources using heuristic algorithms

Virtual Network Embedding (VNE) refers to the instance where multiple virtual networks are hosted on the same substrate network [1]. A substrate network refers to an underlying substance or layer, meaning the underlying physical hardware [4]. For example, if the same infrastructure is used to host virtual networks during the day for commercial businesses as well as virtual networks for residential homes during other hours, it is referred to as virtual network embedding. Virtual network embedding is, therefore, the instance where various network nodes are installed on the same infrastructure; enabling shared capacity between them. When a change in demand is made of the network it is then not necessary to purchase new network nodes; it is merely a matter of changing the software to balance the capacity between nodes dynamically.

1.2 Problem Statement and Motivation

In the network virtualization environment, each VN is described as a collection of virtual nodes that are connected using virtual links to enable communication [1]. Each virtual node is hosted on a particular substrate node in the substrate network, and virtual links span over substrate paths in the substrate network.

Since the node and link resources are virtualized onto a substrate network, multiple VN's that have different characteristics can be created and co-hosted on the same physical hardware. Thus the same physical infrastructure can be used to host more than one virtual network, saving on space and finite resources. The creation of these VNs also creates the ability for network operators to manage and modify networks flexibly and dynamically.

The problem of mapping a VN request with constraints on virtual nodes and links onto specific physical links and nodes in the physical environment which consists of finite resources is commonly referred to as the virtual network embedding (VNE) problem [1], [4]. In order to fully utilise the physical resources that are available the VN requests that are made must be effectively and efficiently embedded since more than one VN is in use of the single substrate physical resources. The benefit that is gained from existing hardware can be maximised through the dynamic mapping of virtual resources onto physical hardware. This optimal resource allocation can be processed considering different factors including, but not limited to, financial profit, energy efficiency and security.

There is an extensive class of combinatorial optimisation problems originating from many complex systems such as:

- radio resource allocation in communication networks;
- network topology management;
- relay resources management of wireless networks;
- portfolio management;
- circuit design;
- mission planning in unmanned systems;
- resource allocation in multi-agent systems;
- reliability optimisation in complex systems; and
- weapon-target assignment in defence-oriented research fields

that are defined as Stochastic Resource Allocation (SRA) problems. Fan et al. [6] put forward that a "distinct feature of SRA is reflected by the fact that SRA solutions depend on the probabilities of stochastic elements or events". An example of this would be that in a defence system the battle effectiveness of the system is dependant on the performance of all the weapon's performance against incoming targets, not just a single weapon. Thus the dependability of a complex system relies on the reliability of all the components within that system. According to the authors literature survey, assorted types of SRA are usually characterised as either a deterministic optimisation problem with fixed probability parameters or as a stochastic one with the certain probability distribution of some key elements.

The purpose of this research will be to possibly optimise the resource allocation of the VNE problem using SRA techniques. Since physical infrastructure is still needed in order to materialise virtual networks the problem of adequate resources to support the optimal network embedding is a growing problem in this mostly technical environment, which is used by people all over the world. The primary purpose of this dissertation is, therefore, to optimally allocate resources for virtual network embedding systems using mathematical models and optimisation techniques and stochastic approaches. This purpose is essential to ensure optimal use of physical resources – that is still used in these virtual systems – in a physical world where space and other resources are not abundant. The end goal is to serve as many clients using the least amount of resources with satisfactory use.

1.3 Research Aims and Objectives

By formulating mathematical models that capture relevant features of the resource allocation process in the VNE context, it is possible to efficiently allocate these resources and produce a solution to these models that are scalable and financially beneficial.

Some of the previous work done in the VNE field split the problem into two separate sub-problems: the node mapping problem and the link mapping problem. These two problems are handled in two different phases. Chowdhury et al. [7] presented a collection of VNE algorithms that leverage better coordination between these two phases.

Sun et al. [8] are quoted: "As a key issue of constructing a virtual network (VN), various state-of-the-art algorithms have been proposed in many research works for addressing the VN mapping problem. However, these traditional works are efficient for mapping VN which with the deterministic amount of network resources required, they even deal with the dynamic resource demand by using over-provisioning. These approaches are not advisable since the network resources are becoming more and more scarce." This is the reason why a stochastic approach is desired in this dissertation. Using a robust mapping algorithm, with a distribution of network requests, the solution can be created in such a way as to enable the embedding algorithm to embed the requests made without over-provisioning and efficiently map requests. This needs to be achieved since physical resources are not abundant and physical space is limited.

Past research does provide a solution for the VN mapping problem but does not provide a scalable, stochastic and optimal solution for stochastic resource allocation.

The objective of using this approach in this dissertation is to solve the VNE problem as an optimisation problem focussing specifically only on stochastic resource allocation with a robust mapping approach. Previous works have formulated the VNE problem in the following ways [9]:

- an un-splittable flow problem
- heuristic-based solution
- meta-heuristic approaches
- exact embedding approaches
- energy consumption of the embedding and other factors

Previous research completed in this field provides a solution to the VNE problem, but these solutions are not optimal and scalable regarding resource allocation when considering other factors as well. To ensure both of the phases of the problem are solved simultaneously, the problem can be considered as a multi-commodity flow constraint problem [9].

Thus, in summary, the following objectives are evident to this dissertation:

- Model the VNE problem with regards to the node and link mapping as a singular problem.
- Use a multi-commodity flow approach when modelling virtual network data.
- Create a MILP model.
- Create an efficient heuristic embedding algorithm for this model.

- Extend the model to accommodate stochastic input requirements
- Test algorithm for scalability and other performance measures.

1.4 Methodology

1.4.1 Literature Study

The literature study for this project will be a comprehensive study on the following topics:

- Importance of Virtual Network Embedding research
- Features of VNE
- Computational complexity of the VNE problem
- Previous research completed in this field
- Mathematical modelling of a system (virtual network formulation)
- Resource allocation considering the VNE problem
- Optimization techniques
- Stochastic and robust approaches (SRA)
- Optimization of the VNE problem

Fischer et al. [1] present a survey of current research in the VNE area. The VNE problem is discussed regarding:

- Static vs Dynamic taxonomy
- Centralized vs Distributed taxonomy

- Concise vs Redundant taxonomy
- Main embedding objectives
- Solutions available for this problem
- Emerging research directions

This paper is a reliable platform for basic information on the VNE problem, what it encompasses and possible research areas.

Tychogiorgos et al. [10] aimed to provide a starting point for researches interested in applying optimisation techniques to the resource allocation problem for current communication networks. This paper concurs that the growing number of applications that require resources in current communication networks emphasises the urgent need for efficient resource allocation mechanisms and argues that optimisation theory can help provide the necessary framework to develop these much-needed mechanisms that can optimally allocate resources efficiently and fairly among the users of the system. This paper features the following topics, in summary:

- Describing key optimisation theory tools needed to create optimal resource allocation algorithms.
- Describing the Network Utility Maximization(NUM) framework that has at present found numerous applications in network optimisation.
- Summarizing recent work in this field of research
- Discussing some remaining research challenges concerning the creation of a complete optimisation-based resource allocation protocol.

1.4.2 Research Design

The proposed design is to model the VNE problem mathematically to represent the system best. This modelling of the problem will determine if the problem is correctly represented. The optimisation algorithm will also be designed using an already established optimisation method (merging more than one method is also possible) in order to create the best solution for optimisation of the mathematical model. This process will take place on a continuous evaluate-and-change basis; meaning outputs will be monitored and evaluated to assess if a change needs to be implemented to the model or optimisation algorithm. This will ensure that the model and algorithm are continuously improving in regards to unforeseen detected oversights that were not foreseen.

In order to obtain results, this dissertation will be looking at two main factors when considering results. Firstly the quality of the solution, is it optimal? Moreover, secondly, the scalability of the solution, to ascertain whether or not the implemented solution can be used on problems of different sizes.

The results will be obtained through the use of case studies. Virtual Network Request (VNR)s are made of the system through the use of request demands, and the resource allocation that ensues will be recorded and measured against different criteria. This comparison is how the effectiveness of the system will be tested. When changes to the mathematical model or optimisation algorithm are made this comparison will be repeated.

The requests that will be made on the system will have a random probability distribution; this is done to achieve a stochastic system response. This robust approach will ensure that the proposed solution will be able to handle more than one particular type of request for a virtual network. This is done to create a solution that can be used in more than one specific situation to create a usable solution.

In order to achieve stochastic allocation, historical data of a user is used to formulate scenario-based demands and decrease resource waste.

The end model will also be evaluated to approximate the change in efficiency and improvement that the model's implements and monetary value will be calculated to represent the change made by the system.

Data Acquisition

Data is needed in the form of virtual network requests. Real world data is preferable, but some assumptions can be made in order to achieve a scalable and time efficient solution. Since no real-world data was available, simulated data were used to test the designed solution.

The data that will be needed are virtual network requests that need resources from the embedded system. This data will include the size of the network, resources needed by that network and other information pertinent to the allocation of resources needed to embed that virtual network.

Data Processing

The data processing used in this dissertation will be of a quantitative nature. Specific numerical data will be available to show the efficiency of the allocation process and the scalability of the solution. This data will not be subject to personal opinions and emotions of human subjects but will be factual data that can be analysed and recreated.

The solution will be evaluated on the following factors:

- Scalability of the solution
- Efficiency of the solution
- Possible financial gain of solution
- Optimality of the resource allocation

1.5 Research Limitations

This dissertation is not focussed on the deeper level of network embedding, but the optimality of the VNE considering SRA. This means that this study is limited in certain aspects including:

- Limiting the study size: to ensure the focus remains on the improvement SRA can achieve.
- Randomized data will be used: as is the case in many other studies since real-world data is not easily attainable.
- Node or link failure is not considered: since this problem is centred on the actual physical implementation of network virtualization, rather than the specific VNE features this study is interested in.
- The virtualization architecture is not considered: since this also focusses on the implementation of NFV, more than the VNE.
- An off-line approach is used: since the specific application this study has in mind falls in line with this approach.

1.6 Dissertation Overview

1.6.1 Chapter Division

The remainder of this dissertation is structured as follow: In chapter 2, the literature surrounding the Virtual Network Embedding problem, such as the factors included in this problem, why it is a problem, what has been done in the field, what resources are needed/used in the problem and computational complexity are provided.

Chapter 3 will focus on optimisation and its techniques. This research will include optimisation as applied to the VNE problem in past research, different optimisation techniques, optimising mathematical models, stochastic resource allocation and other information about this topic.

The mathematical model will be formulated in chapter 4, using the VNE problem as an input. The manner in which the model formulation will be verified and validated will be discussed. Performance measures used in the experiment will be listed and discussed as well.

In chapter 5, the computations for this experiment are documented. The data used and assumptions made with the data will also be discussed and supported. In this chapter, the computational results are critically evaluated, and some broad conclusions will be drawn. The limitations or errors from the model or optimisations techniques used will be discussed, and recommendations for changes can be made.

Finally, in chapter 6, the conclusions that can be drawn from the model or techniques shall be documented and compared to the original hypothesises. Recommendations will be made for future research and changes that can be made to this experiment to make it more realistic for real-world applications.

Chapter 2

Literature Study - The VNE problem

In this chapter the literature is documented regarding the VNE problem, including the importance of performing research in this field, what VNE entails, previous research conducted in this field and the computational complexity of this problem.

The VNE environment comprises of many different factors. The literature study of this dissertation focusses on some of these factors and is summarised in figure 2.1.

2.1 Virtualization

Virtualization technologies enable flexible software design, and this opportunity allows existing networking services that are supported by various network functions connected statically to be transformed into a dynamic and flexible system.

Figure 2.1: VNE Literature

2.1.1 Network Function Virtualization (NFV)

NFV leverages standard servers and virtualization technologies to alternatively being run on purpose-built hardware; network functions can be implemented in software by network operators and service providers. Network functions are transferred from dedicated hardware to general software running on commercial off-the-shelf (COTS) equipment, that is virtual machines, and this is how the separation of hardware network functions from the substrate hardware appliances is achieved. The primary motivator of this technology is that service deployment and testing is becoming increasingly challenging, due to various and fixed proprietary appliances. These are some reasons why NFV was proposed as a critical technology to ensure the proposed gain of virtualization technologies is achieved [11].

NFV is proposed to alleviate certain problems of the traditional network setup [3]. These problems include:

- the rigid and antiquated nature of existing hardware appliances;
- the deficiency of skilled professionals versed in the integration and maintenance of these servers;
- the cost of providing space and energy consumption to a variety of middle-boxes;
- the inflexible characteristics of network service provisioning, and
- the high time to market of new services in the traditional framework.

Figure 2.2 illustrates the NFV framework as discussed in this section [11].

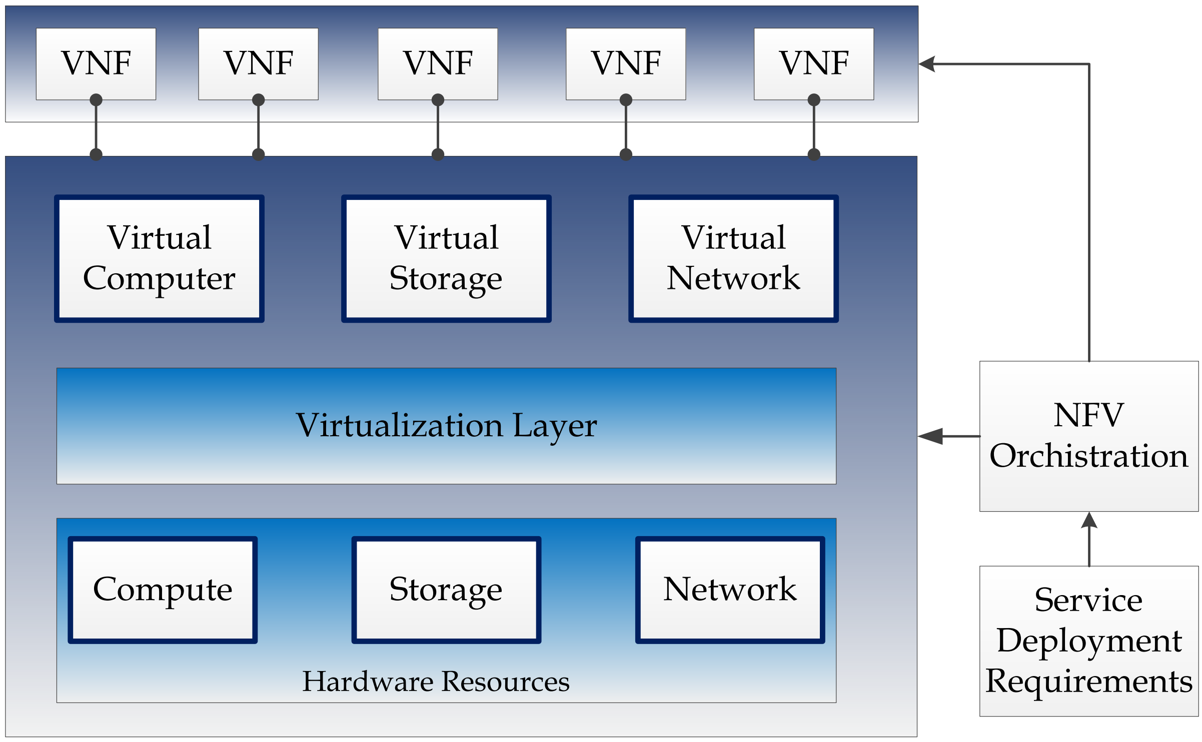

Figure 2.2: Illustration of NFV Framework

NFV decouples the software implementation of network functions from the underlying hardware through the leveraging of full-blown current virtualization technologies as well as commercial off-the-shelf programmable hardware. The IT platforms that can be used by NFV includes general-purpose servers, storage, high-performance switches and so forth. For this reason, NFV is amongst the proposed solutions to alleviate these problems, together with SDN and cloud computing.

NFV fundamentally shifts how a Network Operator (NO) can create and provide their infrastructure through the use of this virtualization technology, this is achieved by separating software instances for its underlying hardware platform and what this achieves is the decoupling of functionality from its location, and this enables faster network service provisioning. This means that the network functions are implemented on commodity hardware through the use of software virtualization techniques. The main advantage of this framework is that these appliances can be instantiated on demand without the need to install new equipment and physical hardware [3].

Li et al. [11] argue that the main area that software-defined NFV can be used is service chaining. Han et al. [3] concurs that a primary concept of this framework is the VNF forwarding graph, which quickly and inexpensively creates, modifies and removes service chains, which simplifies service chain provisioning.

NFV is an essential factor in service chaining, which is an important model for network service providers. A chain dictates data flow order as well as the processes or function required. In order to obtain optimal resource utilisation, these chains must be integrated with service policies. Service chaining is used to organise service function deployment, which supplies the ability to specify a sequential list of service processing for the traffic flows of the service.

Traditional networks mostly rely on manual configurations, which means the manner in which new service requirements are incorporated is time-consuming, complicated, tedious, error-prone and incur a high cost. When a new service requirement is received, new hardware devices need to be deployed, installed and connected in some order. The service provisioning in this framework requires dedicated networking change plans and incurs a high Operational Expenditure (OpEx), which is not desirable. When many diverse service sequences are dedicated to different traffic flows by an operator the OpEx is increased, and the situation is aggravated.

NFV can simplify this service chain deployment and provisioning since it enables less costly and easier service providing in different networks, including local area networks, enterprise networks, data centres and Internet service provider networks [11].

The virtualization technology is utilised by NFV to reduce the Capital Expenditure (CapEx) and OpEx created by various proprietary appliances. Through the use of NFV the ability to deploy different network functions as required in different locations of the network is enabled. The different network locations can be data-centres, network nodes, and the end-node of the network edge etc.

Load balancers, firewall, Deep Packet Inspection (DPI), Intrusion Detection System (IDS), amongst others, are some services offered by hardware dedicated network appliances included in a service chain; this is the setup in the current antiquated networking setup. These services are used to support dedicated networking processing and applications [11].

The authors summarize the commonly used network functions considered for NFV [11], given in table 2.1.

Thus the reduction of the number of middleboxes that must be deployed in traditional networks is a significant advantage when considering NFV, this enables cost savings and provides flexibility to the system. Another factor which is advantageous in NFV is the creation of the ability to substantiate the co-existence of multi-tenancy of network and service functions. This is enabled by the countenance of different services, applications and tenants to use one physical platform [11].

2.1.2 Software Defined Networking (SDN)

In the current architecture of Internet routing decisions and data transmission are both executed by routers, which is why the routing decisions are considered performed in a distributed fashion. The use of this distributed decision making in large networks causes inefficiencies (such as loops and inconsistent routing tables), and it makes the management and control of the network to become very challenging. Interoperability challenges are created because based on different manufacturer's network elements, the control plane (which is in charge of controlling the network elements) will behave differently.

The current distributed architecture, which is based on the concept of Autonomous Systems (AS), has become invalid by the large-scale growth of the Internet.

In recent years there has been a shift in focus to the application of the multi-tenancy concept in the networking domain. The potential for several users to share and use

Table 2.1: Commonly Used Network Functions Considered for NFV

Feature	**Comment**
Network switching elements	i.e., Broadband Network Gateway (BNG), carrier grade NAT, Broadband remote access server (BRAS), and routers
Mobile network devices	i.e., Home Location Register/ Home Subscriber Server (HLR/HSS), Serving GPRS Support NodeMobility Management Entity (SGSNMME), Gateway support node/Packet Data Network Gateway (GGSN/PDN-GW), RNC, NodeB and Evolved Node B (eNodeB)
Tunneling gateway devices	i.e., IPSec/SSL virtual private network gateways.
Traffic analysis elements	i.e., Deep Packet Inspection (DPI), Quality of Experience (QoE) measurement
Next-Generation Networks (NGN) signaling	such as Session Border Controller (SBCs), IP Multimedia Sub-system (IMS)
Application-level optimization devices	i.e., Content Delivery Network (CDNs), load balancers, cache nodes, and application accelerators
Network security devices	i.e.,Firewalls, intrusion detection systems, DOS attack detector, virus scanners, spam protection, etc.
Virtualized home environments	
Service Assurance, Service Level Agreement (SLA) monitoring, Test and Diagnostics	

specific resources as if they were the only customers using these resources is known as the multi-tenancy concept. The shared resources can be physical network elements and links. Although most efforts in this field have been focussed on the application of multi-tenancy concepts to the data centre domain, it is possible to use this concept in other situations.

The owner of the network infrastructure in multi-tenant situations have two identified needs:

- Customer satisfaction - to maximise the Quality of Service (QoS) customer's experience, and
- Monetary gain - to maximise their revenue.

Efficient resource management mechanisms must be implemented in order to meet these needs. SDN has captured attention as a reliable framework to provide support for multitenancy and energy-efficiency because they possess the capability to provide dynamic network management. The control and data plane are decoupled in SDNs, which means that network control and management is centralised and implemented through software while an underlying physical network -compiled by several SDN compliant switches and links- is what the data plane is comprised of [12].

SDN can programmatically configure forwarding rules, and this provides an alternate for traffic steering and is considered an import modern network architecture with the aim to decouple network control from the data forwarding using direct programming [11]. This means that SDN aims to improve the management of a network by centralising the control logic and is commonly referred to as a novel networking paradigm.

This SDN paradigm has different features, summarised below [2, 11]:

- network devices are simple forwarding devices that can be programmed by a SDN controller;
- makes network devices cost less;
- potential enhanced configuration;
- improved performance;
- encourages innovation in network architecture and operations;
- provides a centralised management,and

- the fast reshaping of the traffic through modifications of the flow tables in the forwarding devices.

Thus, since a SDN controller is centralised, it has global information about the network and creates a desirable environment for resource allocation decisions when considering virtualization

Figure 2.3 illustrates the SDN architecture with the three different layers involved [11]. The application layer covers an array of application focusing on network services and is comprised mainly of software applications in communication with the control layer. SDN provides more control of a network through the use of programming by decoupling the control plane from the data plane.

The control layer is considered the core of SDN and is comprised of a centralised controller, which possess the following features:

- a global and dynamic network view is logically maintained;
- receives requests from the application layer and
- manages the network devices via standard protocols.

The data plane layer comprises of infrastructure which in SDN context are programmable and support standard interfaces. The infrastructure can include switches, routers and network appliances.

2.1.3 Network Function Virtualization (NFV) and Software Defined Networking (SDN) Integration

Network ossification and the increase of CapEx and OpEx of service providers is induced through the use of several proprietary network appliances. NFV is the proposed solution to this problem through the enabling of flexible provisioning, deployment,

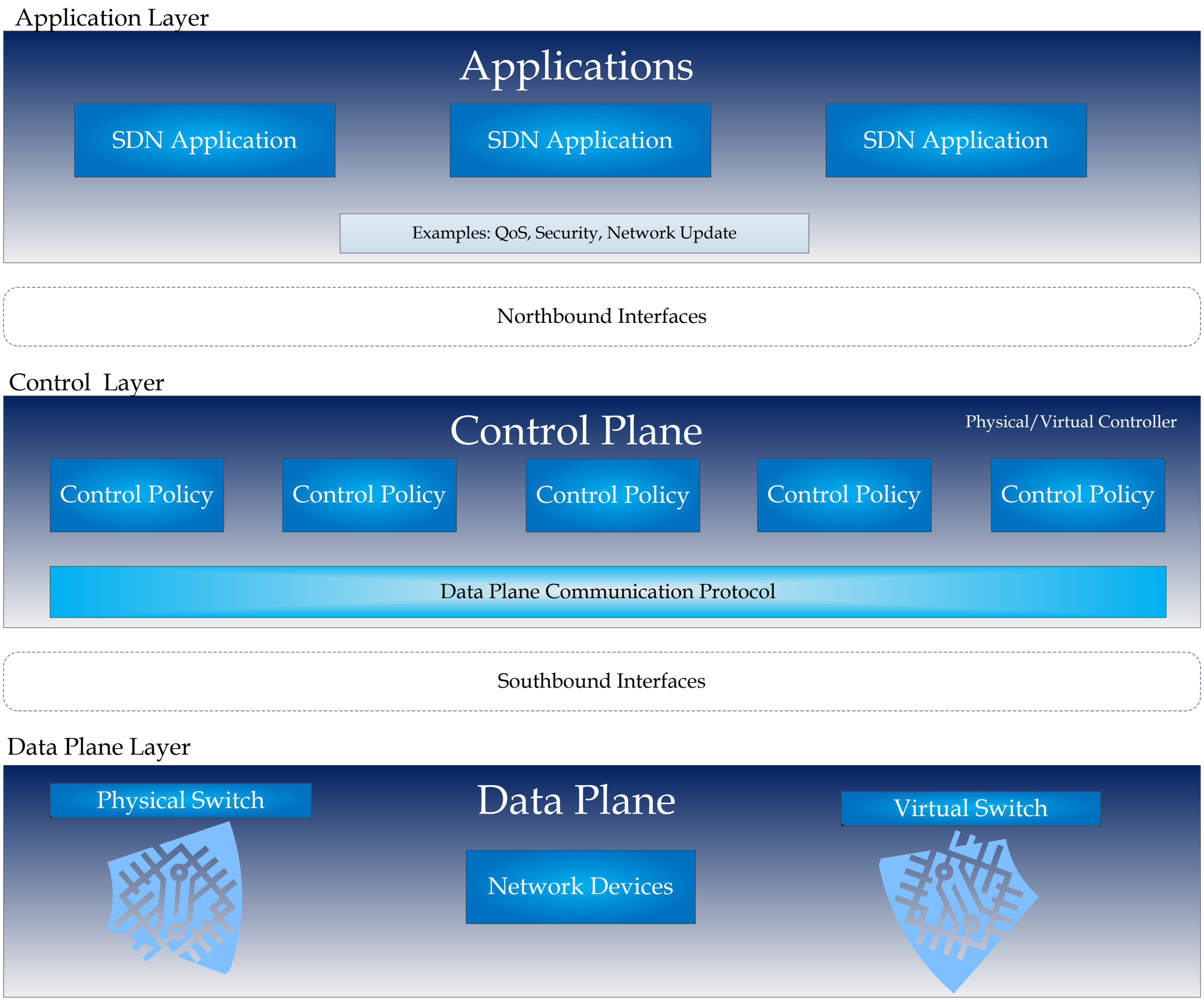

Figure 2.3: Illustration of SDN Architecture

and centralised management of virtual network functions. This architecture can be merged with SDN to create the software-defined NFV architecture. NFV is a good enabler for SDN. NFV offers the capability of dynamic function provisioning, while SDN offers centralised control. This relationship induces new opportunities in service chaining.

This integrated architecture has some additional benefits to pure network function virtualization. These benefits include active traffic steering as well as joint optimisation of network functions and resources. This integrated architecture is becoming the dominant form of NFV since it can benefit a wide range of applications (such as service chaining).

This software-defined NFV architecture integration can achieve improved performance and resource utilisation and is driven by different factors which include the recent trends of:

- the explosion of traffic,
- increased user information demands, and
- diverse service requirements.

The integrated architecture provides the operators with great flexibility, programmability and automation when considering service provisioning and service models [11].

Li et al. [11] states that although NFV and SDN are closely related and high complementary to each other, there are some key differences.

The differences in features of NFV and SDN are summarised in 2.2 below.

Table 2.2: Difference in features of NFV and SDN

Feature	**NFV**	**SDN**
Concept	Implementing network functions in a software environment	Achieve centralized control and programmable network architecture (to provide better connectivity)
Aim	Reduce CapEx, OpEx, physical space and power consumption	Provide network abstractions to enable flexible network control, configuration and fast innovation.
Separation	Decouples network functions from the proprietary hardware (for faster provisioning and deployment)	Decouples network control plane from the data plane forwarding (uses enabling programmability to allow for a central controller)
Motility	moves network functions out of dedicated hardware boxes, to the software (based on general hardware platform)	moves control functions out of the hardware and places it in the software controller

The complementary nature of these two technologies is seen through the fact that NFV can attend to SDN by virtualizing the SDN controller to run on the cloud and through this symbiotic relationship, the dynamic migration of controllers are enabled to move to optimal locations. Likewise, SDN can serve NFV through the provision of programmable network connectivity between virtualized network functions, and this accomplishes optimised traffic steering [11].

2.2 VNE Features

VNE is considered a very challenging resource allocation problem and has been addressed in many different research studies. The main reason for the interest in this field is that the embedding of diverse virtual networks belonging to different users onto a substrate network can maximise the benefits that can be extracted from the infrastructure [13].

2.2.1 Importance of research

The Internet has experienced some noteworthy changes over time. In the past websites were simplistic, static and non-interactive; this has changed to highly dynamic and interactive web applications. The Internet has shifted the platform on which many organisations do business since many businesses now perform online and remain in touch with their clientèle. When this first started, there were some significant challenges to be resolved, mainly concerning bandwidth and capacity, which led to much research being completed on bandwidth sharing and congestion avoidance. Recently there have been noticeable improvements in networking technology which has to lead to an increase in bandwidth. However, this increase in available bandwidth has caused Internet users to - in general - increase their consumption of bandwidth.

Currently, the Internet is utilised as a platform for a wide range of interactive services since real-time communication applications have evolved synchronously with the increased bandwidth trend. These interactive services include media such as chat, remote desktop, stock trade systems, internet banking, IP telephony, video conferencing and network games. This interactivity has a detrimental effect, however since latency requirements arise and users become dissatisfied when they must wait for a system to respond to their demands.

This can create an awkward situation since the underlying architecture of internetworking is usually based on best-effort services. Providing a constant QoS of data transport has gathered some research but not in a significant manner due to most successful approaches needing support along the path of connection. This reliance on end-to-end approaches to providing data is a consequence of the lack of QoS mechanisms [14].

In recent networking literature, there is a prominent topic, network virtualization techniques. A generalised idea of this epitome is the decoupling of the high-level role of service provisioning from the low level one of management and operation of the Substrate Network (SN); this means the software implementation of network functions are separated from the underlying hardware. This decoupling creates a more flexible situation which can aid in the prevention of Internet ossification - or at least partially - since this ossification is a consequence of the difficulty in upgrading physical topology of large, pre-existing networks [3, 5].

For these reasons, the future Internet is seen as being enabled by network virtualization, because this technology has the goal to overcome the current Internet opposition to structural modification [1].

The main advantages of sharing of physical network infrastructures are, amongst others,

- high energy efficiency;
- improved flexibility;
- improved manageability;

- improved scalability, and a
- chance of robustness for mobile operators

These advantages are well documented in the literature. These listed advantages are not inclusive of the increased utilisation of the available physical resources since this is the main motivator of this approach [15].

Resource Allocation Problem (RAP)s are a common problem prevalent in many mission-critical systems such as smart grids and are a problem dealing with the balancing of supply and demand [16].

Current communication networks have a problematic issue because the increasing number of applications that are created are all competing for resources of current networks. This competition between users of networks for resources showcases the necessity for efficient resource allocation mechanisms to maximise user satisfaction [10].

The algorithms - referred to as VNE algorithms - that are an application of this technology is responsible for the initiation of virtualized networks onto substrate infrastructure while optimising the layout by service relevant metrics [1].

In real-world situations, it is sensible to assume that the actual demand of computing resources and traffic - for each VN - can vary over time, and often significantly. An example of this is online gaming or video streaming service that can have more or fewer customers at a specific time and as a result, can have different resource consumption which will depend on its popularity. A service's popularity changes over time, such as a peak when new content is released or the completion of an advertising campaign. These fluctuations in demand create a network reliability issue since it can lead to traffic congestion, QoS degradations, as well as service disruptions.

The previous solution to this problem of data uncertainty was to usually consider only a maximum setting, referred to as a worst case setting. This approach was used to guarantee network operation, even for peak traffic situations. Furthermore, the feasibility guaranteed by this approach comes at a price; an increase of unnecessary costs and in many cases it is highly unlikely for a VN to reach peak demands for all its demands simultaneously.

This is the reason that it is considered reasonable to assume that the probability is relatively small that all demands will simultaneously reach their peak values in various real-world cases. Referring to the example mentioned earlier, it is like assuming that the advertising campaign and the release of new content will not overlap for all services simultaneously.

The goal of all of this is to seek a solution where different VNs are provisioned for demands that are less than their peak values. This will guarantee that the SN has adequate capacity for almost all traffic configurations and only neglect a few extraordinary cases. Furthermore, this approach will potentially obtain more profitable solutions, in which more VNRs are embedded, and issues of over-provisioning are decreased [17].

Sun et al. [8] concurs that network resources are a scarce commodity and should be used accordingly, not wastefully allocated and over-provisioned to virtual network requests. This paper argues that the approach used to map resources in the VNE problem efficiently should use a stochastic approach, which is superior to traditional solutions. This is an example of how the RAP can use a robust approach.

2.2.2 VNE Environment

VNE strategies are moving from theoretical to real scenarios and attempting to solve its specific problems. Two categories that are prevalent for these sophisticated network virtualization techniques are wireless and optical networks [1].

Wireless networks

Currently, wireless networks are the leading type of access technology, and virtualization is expected to be applied in wireless scenarios as well. Wireless links have a propagated nature, and therefore the main distinctive feature of wireless network virtualization is how to virtualize links. Link virtualization has been completed with time-division multiplexing, frequency-division multiplexing and space division multiplexing. In VNE for wireless networks the most challenging problem to face is the nature of propagation /broadcasting of wireless links and this can cause interference with other wireless links.

Existing approaches are considered to miss some important features which would be paramount to wireless network environments, such as mobility and distribution, that must be found in future works aimed at wireless networks.

Optical networks
The Virtual Optical Network (VON) mapping concept was proposed in previous works, as such the VNE problem in optical environments is defined as the maximisation of the number of mapped VONs from the demand set given the limited capacity of the optical SN. There are two different versions of this problem proposed in previous works, which are:

Transparent optical mapping
Optically transparent end-to-end services are provisioned over the VON. When referring to transparent services, it merely means that the optical VN is not assumed to have electronic termination capabilities in nodes and furthermore VONs have to allocate the same set of wavelengths for every virtual link.

Opaque optical mapping
In this mapping structure, the optical VN is assumed to have electronic capabilities, and this means that the same set of wavelengths for each virtual link does not need to be allocated. These problems are formulated and solved using interlinear programming techniques.

Industry and academia have both focussed much attention on VNE, and existing studies can be broadly classified into two types of VNE: VNE for Internet and VNE for data centres. These classifications are discussed below [18]:

VNE for Internet

Simulated tempering has in the past been introduced to manage the NP-completeness of the VNE problem by randomly generating a nominated mapping solution and algorithms improved this nominated solution through iterative adjustments.

Computation and network resources are not the only factors that affect the quality of embedded solutions; topology can also be an essential factor.

Topologies for VNs have the probability to be rather substantial, due to latency and complexity reasons it may not be feasible to map such a large VN. A solution can be to divide a VN into an array of smaller, more basic clusters.

VNE for Datacenters

Cloud tenants commonly care to possess the ability to predict the performance of their applications that are placed in data centres. This accurate prediction ability is not genuinely feasible due to time-varying workloads of applications and resource contention in production datacenters

Liang et al. [18] note that existing studies on VNE mainly put their attentions on:

- finding coordinated node and link embedding;
- inter-domain embedding;
- physical node load balancing;
- better metrics for physical nodes and
- large-scale embedding

Amongst other things. The authors also state that most of these previous works did not consider the possibility of splitting a virtual node into multiple smaller ones. This splitting of virtual nodes can potentially improve physical resource utilisation and increase virtual network requests acceptance ratio.

The authors in [18] noted their article contributions as

- introducing the idea of embedding parallelizable virtual networks by splitting each virtual node into multiple smaller ones, which is claimed to improve the resource utilisation;
- presenting two embedding algorithms that proactively - or lazily - utilise parallelisation and
- providing three extensions to complement their proposed algorithm.

2.2.3 Problem

The Internet is becoming increasingly ossified since its exponential growth encourages the development of new technologies and applications, but its large-scale hinder the deployment of such new features. Furthermore, there are various Service Provider (SP)s which means that when new architecture must be applied, many mutual agreements must be made among the Internet Service Provider (ISP)s and this demands changes in the routers and central computers. An approach to solve this problem is proposed through network virtualization, where the physical - or substrate - network provider provides a physical - or substrate - network to support virtual networks. This approach ensures that there is no need to change the physical network or negotiate contracts between the ISPs when new technologies are deployed [19]. The supplier framework for this problem is discussed in section 2.6.

Network virtualization has a dynamic and programmable network environment, which means that this technology can improve the flexibility of the current network architecture, promote network innovation, and address the Internet ossification problem [20].

The primary entity of network virtualization is the VN, which is a combination of active and passive network elements (network nodes and network links) hosted on the physical SN. In a VN virtual nodes are interconnected through virtual links, and this creates a virtual topology. Various VNs are isolated from each other and can be deployed for customised end-to-end services for end users [1,20].

Various VN topologies that have widely diverging characteristics can be created and co-hosted on the same physical hardware when the node and link resources of a SN are virtualized. This abstraction created by resource virtualization mechanisms permits NOs to manage and modify networks in a highly flexible and dynamic way [1]. These various VNs coexist on the same SN, which has finite resources and these are shared among the VNs [20].

Figure 2.4 [21] shows a typical network virtualization environment.

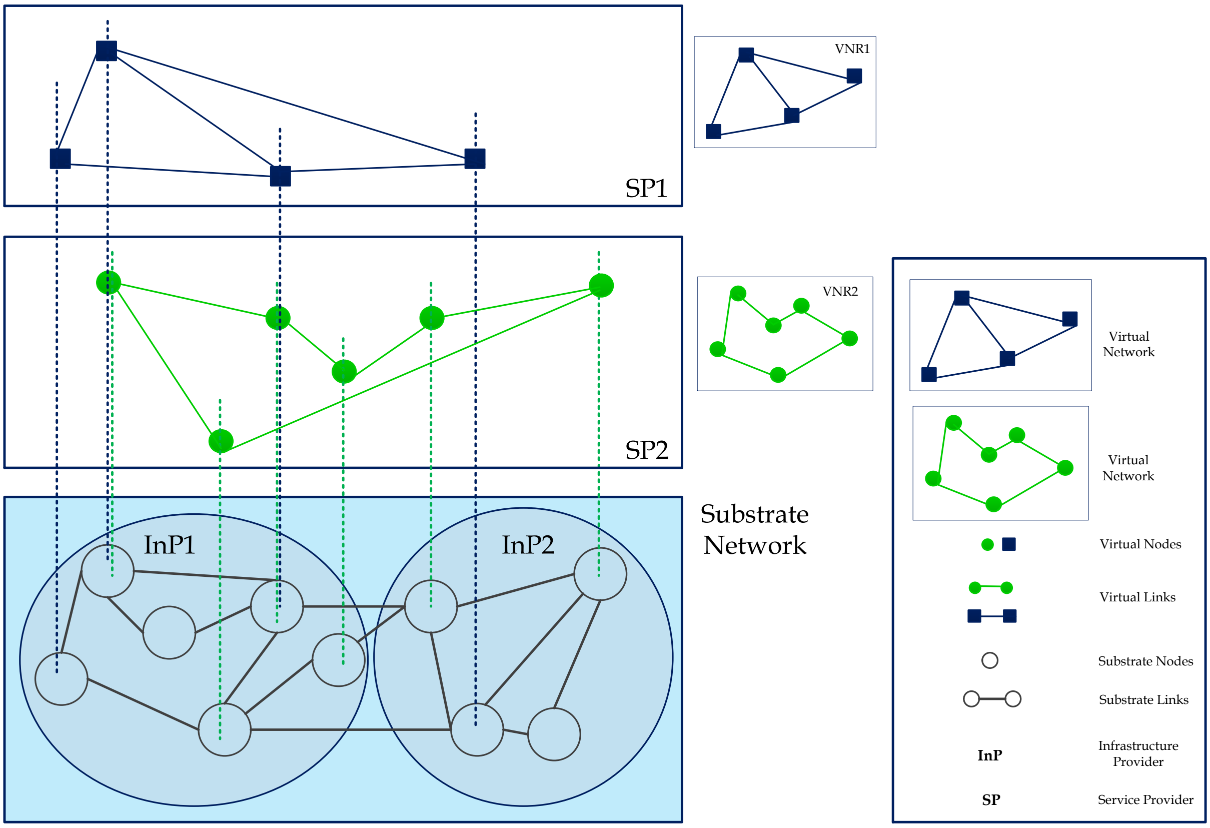

Figure 2.4: Network Virtualization Environment

In a scenario, there will be a SN and a group of VNs. The group of VNs are referred to as VNRs, where each request consists of a set of resources that are required by the demanded service. Thus the VNs must be optimally allocated over the SN since the primary objective of VNE is the optimal allocation of these VNRs onto the SN based on some predetermined objective. In the phase of the problem where nodes are mapped, each virtual node of a VNR is mapped onto a substrate node that contains enough capacity to satisfy that virtual node resource demand. VNRs will consume CPU processing power on nodes of the SN. In the phase where virtual links are mapped, each virtual link is mapped to a directed path in the SN that has enough resource capacity to fulfil the virtual link resource demand. VNRs will consume bandwidth capacity on links of the SN. The limit of VNE is presented by physical resources since they are finite [21].

VNE is discussed below [2,4,5,7,17–19,22,23]:

- VNE is also referred to as Virtual Network Assignment or virtual network mapping problem
- Is a significant problem in the domain of network virtualization and problems of resource allocation.
- It aims to optimise specific objective functions (such as revenue or energy efficiency)
- There is a substrate - or physical - network SN which has a node and edge capacities; these resources are finite.
- There is a set of virtual networks VNs which have node capacity demands and link (node-to-node) traffic demands.
- Each VN is merely a collection of virtual nodes connected by a set of virtual links.
- Resource demand (of the vertices and edges) of the VN must be less than that available with a SN.
- In a VN a virtual node is hosted on a particular substrate node. Furthermore, a virtual link spans over a path in the substrate network.
- VNs and SNs can be represented by a graph in which vertices of the graph depict the network nodes, and the edges depict the network links.

- VNE must embed these VNs onto the specific nodes and paths in the SN in an optimal manner.
- The embedding of the multiple VNs in a shared substrate deals with the efficient mapping of virtual resources in the physical infrastructure and is referred to as VNE.
- This embedding must be performed efficiently and effectively since multiple heterogeneous VNs shares the underlying physical resources, which poses a significant problem.
- VNE maps these VNs to SNs subject to certain resource constraints (such as CPU or bandwidth requirements).
- VNE has been proven to be NP-complete.
- VNE is aimed to maximise total profit - profit function - while adhering to the physical node and edge capacities.
- VNE must decide which VNRs that are issued by customers to accept or to reject referred to as admission control.
- VNE must decide how to allocate the physical resources to the accepted VNs .

The VNE problem is currently a well-studied problem with multiple application domains. In figure 2.5 [21] the following is illustrated:

- One SN with four nodes.
  Substrate nodes can host several virtual nodes (up to two in this example). Likewise, substrate links can host more than one virtual link.
- Two VNRs with three nodes each need to be hosted.
  One of the virtual links spans two substrate links, thus representing a virtual resource combined from several substrate resources.

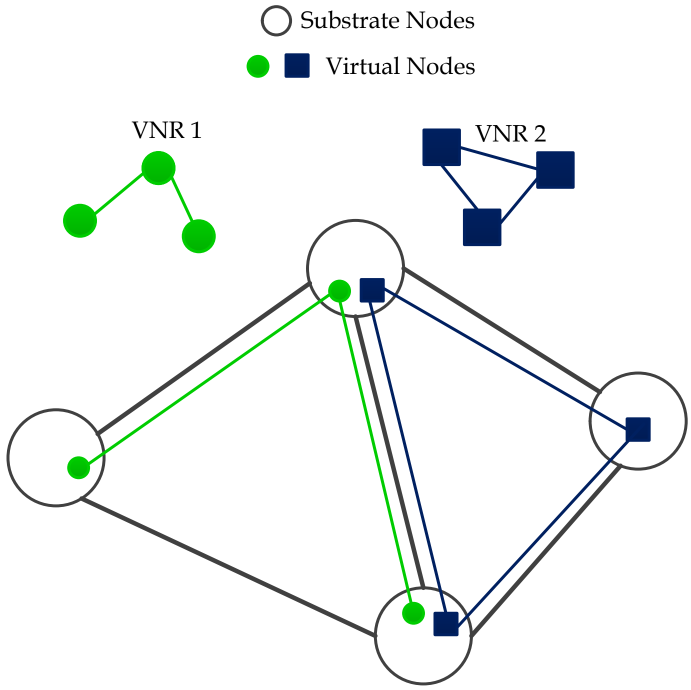

Figure 2.5: Embedding of Virtual Network Requests

The assignment of virtual private networks (VPNs) in a shared provider topology and the network test-bed mapping problem are closely related to the VNE problem [1].

Fisher et al. [1] states that future Internet architectures will be based on the Infrastructure as a Service (IaaS) business model that decouples the role of current ISPs into two new roles [1] and is comprehensively discussed in section 2.6. VNE approaches can be categorised according to whether they are Static or Dynamic, Centralized or Distributed, and Concise or Redundant. These categorisations will be discussed in section 2.3.

2.2.4 Problem Decomposition and Coordination

The VNE problem can be separated into two distinct phases, also referred to the two subproblems of VNE [1,7,24,25]:

1. Virtual Node Mapping (VNoM)

   The virtual node mapping stage is where virtual nodes - that contain computational capacity requirements are assigned/allocated to the substrate - or physical - nodes. Some existing algorithms assign virtual nodes using greedy heuristics, such as assigning virtual nodes that have higher requirements to substrate nodes that have more available resources.

2. Virtual Link Mapping (VLiM)
   The virtual link mapping stage is where virtual links - that have bandwidth capacity requirements and connect virtual nodes are mapped onto substrate links/paths. This mapping connects the corresponding substrate nodes after the node mapping stage. Some existing algorithms are embedding virtual links onto substrate paths using shortest path algorithms when considering unsplittable flows; otherwise, for splittable flows, multicommodity flow algorithms are used.

The allocation of resources must be completed optimally to improve resource utilisation, altogether while fulfilling both node and link constraints in the VN [24]. These two subproblems of VNE must be solved in order to solve the VNE problem. Alternatively to VNoM/VLiM coordination, it is possible to solve each subproblem in an independent and isolated manner. In this setup, VNoM must first be solved before VLiM since the former's output provides input for the latter. When the problem is solved using independent phases, it is referred to as uncoordinated VNE [1].

Uncoordinated VNE

In this approach, the solution is achieved in two different stages since there is a lack of coordination between the VNoM and VLiM. The VNoM is solved as the first stage, and the output of this stage is used to solve the VLiM as the second stage of the approach. In older works, this approach is widely used, and the objective of these works was to maximise the long-term average revenue. There is, however, a problem when there is a lack of coordination between the node and link mapping since this separation can result in adjacent virtual nodes being widely separated in the substrate topology. This problem increases the cost of single/multi-paths that are used to achieve a solution to the virtual link mapping phase and leads to low acceptance ratio and thus decreased long-term revenue.

First stage - VNoM
The VNoM in the first phase of achieving a solution to the VNE problem uses greedy algorithms, and these algorithms choose a set of available substrate nodes for each virtual node and so will assign one of these choices based on the available resources. The goal of this phase is to assign virtual nodes with larger demands to substrate nodes with more resources.

Second Stage - VLiM
The VLiM phase is solved in two separate manners which depend on the assumptions that were taken for the SN. The first is referred to as single path mapping using one k-shortest path solution for increasing k, and this approach is used when each virtual link has to be mapped to only a single path in the SN.

The second approach is multiple path mapping and is used in the case where each virtual link demand can be transmitted by several paths in the SN. In this approach, VLiM is reduced to the Multi-commodity Flow Problem (MCF) problem which allows a multi-path routing solution for each virtual link by using optimal linear programming algorithms.

Coordinated VNE

From this information, it is seen that coordination between the node and link mapping is desired in VNE approaches. When VNoM is completed with no consideration of its relation to link mapping, then the solution space is restricted, and there is a decrease in the overall performance of the embedding. Coordinated VNE can be achieved in two manners: in two or one stage.

Two stages coordinated VNE
This approach has the main goal of minimising embedding cost. Geographical location for substrate and virtual nodes and a non-negative distance per VNR indicating how far a virtual node of the VNR can be of its demanded location are new sets of node

constraints that are added.

The node mapping phase commences with the creation of an augmented graph over the SN, which introduces a set of meta-nodes (one per virtual node), each connected to a cluster of candidate SN nodes adhering to location and capacity constraints. Over this augmented graph, the algorithm solves the VNoM problem by proposing a Mixed Integer Programming (MIP) formulation. The primary goal of the MIP formulation is to solve the VNE while trying to minimise the embedding cost, considering that the added meta-nodes map the requested virtual nodes.

MIP can be NP-complete, and in order to avoid this, the linear programming relaxation solution is achieved and virtual nodes are mapped to real substrate nodes and not meta-nodes. This is achieved by rounding the obtained solution in two different ways, deterministically or randomly. These two approaches result in different solution algorithms (DViNE and RViNE).

After completion of this phase, the VLiM is completed. The MIP formulation that is used for the virtual node mapping also takes into account the mapping of virtual links in the augmented SN between the source and destination meta-nodes. Thus the coordination between the two phases is strong and selected substrate nodes that will host the VNR nodes are more probable to provide virtual link mappings with low embedding cost.

One stage coordinated VNE
In this approach, the virtual links and virtual nodes are mapped at the same time. The approach works by mapping the virtual link between a virtual node pair as the first virtual node pair is mapped, as well as when each virtual node is mapped the virtual links connecting it with already mapped virtual nodes are also mapped.

The efficiency of the embedding is directly influenced by the topological attributes of a node. The PageRank algorithm used by Google's search engine inspired a node ranking approach that is proposed to measure the topology incidence of a node. There is an increase in acceptance ratio and the link mapping efficiency when topology attributes are incorporated in the node mapping. When considering two nodes that are equal in resources, the node with the more capable neighbourhood will be chosen, leading to a higher success probability for the embedding, and this is the reason for the increase. Figure 2.6 and 2.7 [1] illustrates how this approach functions.

The algorithm proposed in previous works (RW-BFS) provides a solution to one stage VNE and is based on the breadth-first search (BFS) approach. Since this is a one stage solution, the proposed algorithm takes into account the effect of the node mapping on the link mapping stage.

The first phase of the RW-BFS is to calculate the node ranks for substrate and virtual nodes, and this is shown in figure 2.6. From this phase, the algorithm creates a BFS tree of the VNR, in which the root node is the virtual node with the highest rank and the children are placed from left to right based on their rank. The VNE is then completed by mapping each virtual node in the first feasible substrate node of its list moving along the BFS tree. Furthermore, simultaneously the virtual link incident of that virtual node is mapped on the substrate shortest path that fulfils the bandwidth demands.

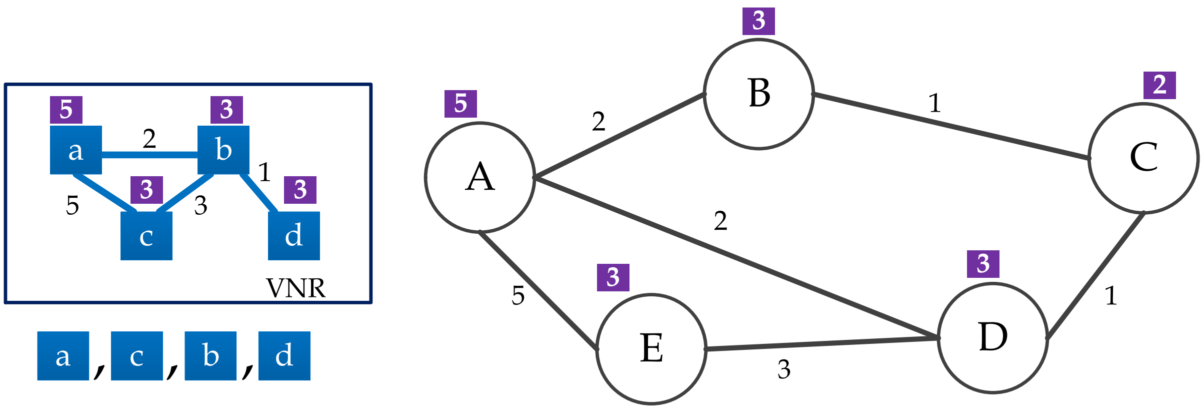

Figure 2.6: Illustration of One Stage Coordination - A

An illustration of how this one stage VNE is performed is given in figure 2.7. In this figure, there are certain intervals, and the corresponding actions are as follow: At t1 and t2 the virtual nodes labelled as a and c are mapped onto the substrate nodes A and E respectively. At t3 the virtual link between the already mapped nodes of a and c is mapped. The subsequent virtual nodes and links are mapped following the same procedure, combining, in this way, the virtual node and link mapping in one single stage.

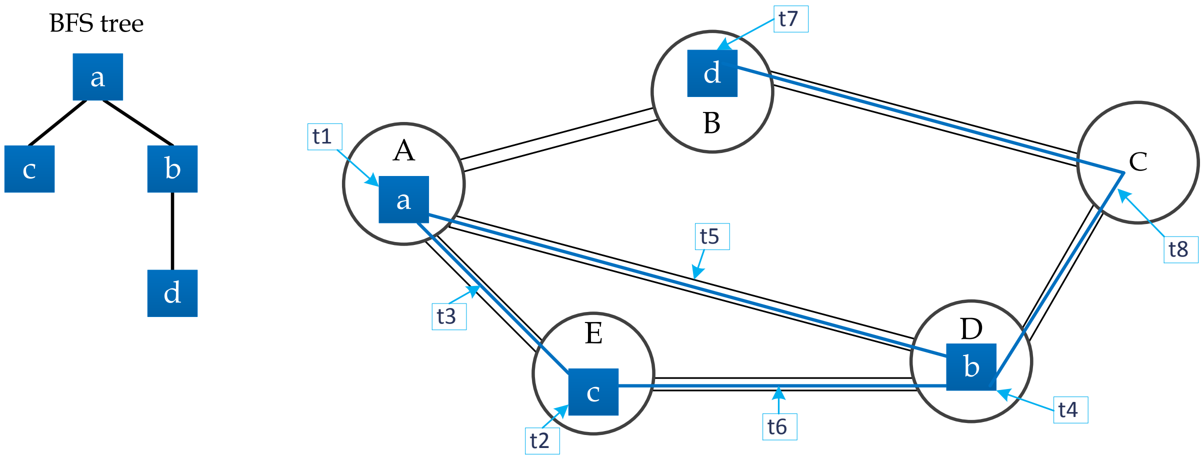

Figure 2.7: Illustration of One Stage Coordination - B

InterInP Coordination

Some versions of VNE solutions consider the VNE with only a single Infrastructure Provider (InP), which is referred to as IntraInP VNE. This may not be realistic since practically VNRs will be mapped on top of a set of SNs managed by different InPs. In this scenario, it is noted that each InP should be able to embed parts of the virtual network and connect them using the external links among InPs.

In this setup, each InP provider must be able to embed certain parts of the VN, as well as use external links to connect them among the InPs. Furthermore, to minimise the embedding cost the SP divides the VNR into various sub-requests, and each of these sub-requests is mapped in the SN that is the most convenient. Individual InPs map these sub-requests using common VNE algorithms. This concept is illustrated in figure 2.8 [1].

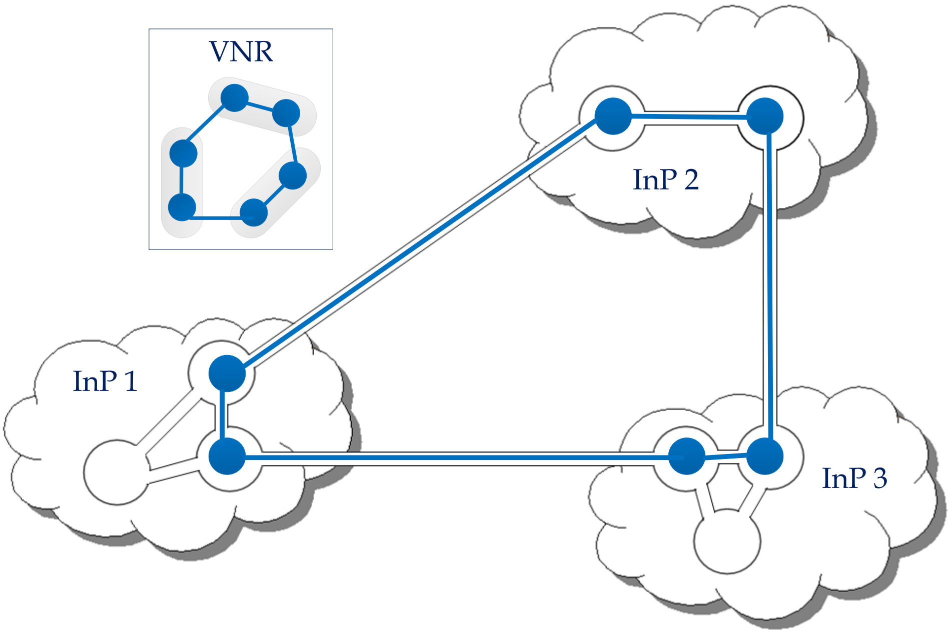

Figure 2.8: Illustration of InterInP

Some works represent this problem in a suitable manner, where the authors address some conflicts of interest between SPs and InPs. The SP aims to satisfy its requirements while still minimising expenditure. On the other hand, the InPs aims to optimise equipment allocation; this is achieved by acquiring requests specifically for their high margin equipment while offloading unprofitable work onto their competitors. The goals in these approaches are thus to use a distributed protocol that will coordinate the InPs, and this ensures competitive embedding pricing for the SPs

2.2.5 Parameters

The resources used in the VNE problem are attributed to specific parameters. Firstly, there are substrate resources that have their capacities and qualities, for example, a substrate node can supply a particular computing capacity which is related to the CPU available to it. Secondly, virtual resources each have their respective requirements; for example, a virtual node will require a specific computing capacity to compute routing information accurately. These parameters are considered extremely important to achieve valid embedding, and the different kinds of metrics are discussed below.

- Linear parameters.
  Linear parameters are necessary when using Linear Programming (LP) techniques to create optimal solutions.
- Nonlinear parameters.
  When non-linear parameters such as path loss probability are used, LP cannot be used.

There are also various other dimensions into which parameters can be categorised. As an elementary distinction there are:

- Node parameters.
  These are the attributes that refer to nodes, like computing power.
- Link parameters.
  These are the attributes that refer to links, like bandwidth.

This strict model does, however, create a problem. In the case where a virtual link is mapped to a path in the SN, the computing power of substrate nodes on the path may have an impact on the virtual link bandwidth. Virtual link consist of a combination of substrate node and links and virtual nodes only considers substrate nodes, for this reason, it is considered slightly easier to map virtual nodes than to map virtual links

Considering this information, there is another further distinction that can be made:

- Primary parameters.
  These parameters can be directly assigned to a substrate resource and, likewise, can be explicitly required by a virtual network request.

- Secondary parameters.
  These parameters depend on other (primary) parameters, for example the probability of packet loss at a node depends both on its computing power and the size of its memory.

During embedding it is easier to consider primary parameters - since they can be directly matched - but not secondary parameters since they must be calculated first. For links, such as wireless links, packet loss can also be a feature, and in this case, there can exist another distinction between consumable and static resources.

- Consumable resources.
  These are resources that are consumed when a virtual entity is mapped, such as computing power and bandwidth. The variability of bandwidth demanded - in a mapped virtual link - is taken into consideration in some VNE approaches by either considering it as a stochastic variable or by opportunistically sharing it among different flows.
- Static resources.
  These resources are not dependant on the number of virtual resources mapped to a substrate resource. For example, the loss probability of a wireless link in the SN will stay constant, no matter how many virtual links are mapped to it, provided the bandwidth - which is a consumable resource - is not exhausted.

Static resources are typically easily matched during embedding, while consumable resources usually induce the NP-completeness of the embedding problem. There is yet another distinction that can be made:

- Functional parameters.
  These parameters stipulate low-level functionality, e.x. computing power and bandwidth.

- Non-functional parameters.
  These parameters are high-level properties of the respective entities, e.x. security or resilience of an entity.

Even though it can be difficult to compute, functional parameter matching is usually straightforward and non-functional parameters typically require more elaborate approaches in an embedding algorithm [1].

2.2.6 Resource Allocation

The goal of a RAP is to specify the contributions of system components in such a manner that their sum satisfies an established demand. There are various technical systems where this problem is a subject of importance, for example in the gas pipeline and water supply systems the challenge is to maintain the pressure of the system at a certain level [16].

Several performance metrics can be used for the VNE problem and are discussed in section 2.4.3. The choice of parameters used differs in most research works, for example, Chowdhury et al. [7] used five metrics in their evaluation to measure the performance of their algorithms against the existing ones, which included acceptance ratio, generated revenue, provisioning cost, average node utilisation and average link utilisation. There are many metrics, and these will be discussed later in this dissertation, as mentioned.

Scheduling problems have two central and interconnected challenges when specifically considering large-scale systems [16]:

- Scalability.
  Solving a scheduling problem is NP-hard, both about the number of agents involved and time steps.
- Uncertainty.

Even though there are uncertainties, RAP needs to be solved. The uncertainties are produced by specific circumstances such as inaccurate demand predictions as well as by the system participants themselves.

Certain challenges must be considered to achieve proper CPU, memory, and disk I/O resource allocation, as well as efficiently providing adequate application performance guarantees. These challenges include [26]:

1. Automation
   A virtual machine monitor(VMM) is responsible for global resource allocation, and an individual Virtual Machine (VM) is responsible for its application in a highly autonomic manner in virtualized autonomic systems. When many VM's resource demands change, a reconfiguration of one or more VM allocation may be required to guarantee system performance and Service Level Agreements(SLAs) based requirements. In reaction to the frequently changing data centre conditions, it is also desirable to allocate resources among various VMs while still satisfying their service level agreement requirements. This is why it is contended that allocation decisions must be made automatically and without human intervention, not manually to provide real-time performance assurance.

2. Adaptation
   The distributed nature of multi-layered virtualized environments makes the resource allocation - that is based on operating systems with full knowledge and full control over the underlying hardware resources - approaches inadequate. Since this scenario can make it difficult - and sometimes impossible - to characterise the workload the embedding must be adaptable considering workload variations. This means the workload characteristics should not influence system conditions to such a major extent.

3. Scalability
   Contentions and conflicts are inescapable in a highly amalgamated environment where VMs share multiple resources at the same time. Resource allocation archi-

tecture - without requiring a single centralised controller or hardware capabilities - must have the ability to scale to a large scale virtualized environment with many applications and physical nodes. These goals cause that modification is needed when applying traditional resource allocation techniques to the virtualization environment since they can no longer be used directly. The problem then becomes to allocate appropriate resource to the upper-level services and applications that have contending even conflicting resource demands.

2.3 VNE Classifications

2.3.1 Online vs Offline

One classification of approaches to the VNE problem is defined by either having an online or offline version of the VNE problem. Both of these versions are considered NP-hard and most research in the field do not expect to be able to efficiently compute an optimal solution [7].

Most authors concur that the offline version is used when the VN requests are known in advance [13, 17], although some add the assumption that the substrate network has unlimited resources and can accept and handle all of them [22] or that no admission control is executed when there are insufficient physical resources [24]

Amaldi et al. [5] completed an offline version approach to the problem through the assumption that not only the virtual but physical networks as well are undirected graphs and used unsplittable single path routing. This approach is, in many cases, favoured when considering applications, since it avoids the problem of packet reordering.

Likewise, the problem is considered as the online version when VN requests are dynamically received by the system over time. Thus the on-demand virtual network assignment problem is analysed, and a VNE algorithm dynamically allocates resources and embedded virtual network requests at the arrival of each request [5, 13].

Some works argue that the online version is more inflexible, which is apparent given that the VNE problem is NP-hard, even considering the offline version of the problem. After the virtual nodes are already embedded, the problem of embedding the virtual links is still NP-hard as its problem [13].

The online stochastic optimisation is sometimes used to combat the uncertainties of different sources when explicitly considering the field of operations research [16].

Chowdhury et al. [7] even argues that considering a current objective is to increase the utilisation of underlying resources and InP revenue, the version of this problem which is of the most important is the online version.

Some research on the online version of the problem produces a remote node and link mapping since they consider the local resource of nodes in the VNE problem [24]. Some works argue that in real world situations it is not realistic to know all the VN request beforehand. These works concur that in the real-world application online algorithms are preferred and more suitable for the situation [13].

This is true when considering some real-world applications, but may not be true in specific versions of the problem. Consider a SP that rents out the virtual network by the hour, in this scenario it is advisable to use the offline version of the VNE problem since the request will be known before the embedding must take place. Thus the specific situation will influence the problem version that is most suited.

2.3.2 Centralised vs Distributed

VNE algorithms can be classified into centralised algorithms and distributed algorithms, considering their control mechanisms [22]. These two approaches are fundamentally different and possess their advantages and disadvantages [1].

The centralised approach is summarised below. Description of centralised approach [1, 12, 22, 23, 27] :

- Has a single, centralised entity responsible for performing the embedding;
- Control entity has a global substrate view ;
- Global view avoids conflicts and inconsistencies during the mapping process;
- Entity can be a dedicated machine computing optimal solutions;
- Single control entity receives VN requests;
- Central coordinator maintains global information about substrate network, and
- Several centralised approaches have been proposed in the literature to provide optimal resource allocation schemes.

The advantages and disadvantages of the centralised approach is given in table 2.3 [1, 12, 22, 27].

Table 2.3: Advantages and Disadvantages of a Centralised Approach

Centralised Approach	
Advantage	**Disadvantage**
Central entity responsible for the embedding is aware of the global situation of the network	Central control entity needs to maintain up-to-date global information and manage every action of the substrate network, and this requires efficient message-passing mechanisms. This can be considered the hot spot of the system.
Because the mapping entity is at every step of the mapping, it is aware of the overall situation of the network, and this can facilitate more optimal embedding	When the substrate network is dynamic and on a large scale, maintaining the global status information of the entire substrate network becomes an enormous challenge. For this reason, these centralised algorithms sometimes suffer from high latency and do not scale well for large topologies
The global information view of this approach aid in the avoidance of conflicts and inconsistencies in the system and thus can easily obtain globally optimal solutions (or at least near-optimal solutions) to map resources for corresponding VNs	If the centralised entity fails, the entire mapping process fails, since the control entity presents a single point of failure. Centralised approaches also need perfect information and cooperation among the centralised entity and the other agents in the network, such as the controllers. Because controllers are unlikely to share their private information, centralised solutions are hard to implement in competitive scenarios where information is neither revealed nor shared among parties. Controllers tend to act selfishly and seek to maximise their profits unilaterally. Thus cooperation is not always possible.
	The amount of information that must be gathered in a centralised approach incurs many signalling messages, and this creates a bottleneck when the number of VN requests increases. For this reason, this approach limits scalability and increases complexity in highly dynamic or large scale virtual network embedding environments; furthermore, it often leads to an NP-hard problem that can only be implemented in real-world systems if sub-optimal solutions and performance losses are tolerable.

The distributed approach is summarised below. Description od distributed aproach [1,22,23,27]:

- Distributed approaches utilise multiple entities to handle embeddings or have multiple entities taking part in the decision making for the embedding;
- Distributed algorithms complete the mapping process in a decentralised manner;
- The organisation of how mapping is distributed among the participating entities can either be internal, or completely ad-hoc;
- This approach offers better scalability since the load is distributed among several nodes;
- This approach aims to alleviate the scalability and robustness problems of centralised algorithms;
- There are less of these approaches used in current works ;
- Since there is no central control entity this approach relies on a great amount of message exchange among substrate nodes to achieve the VNE, and
- There exists a special version of this approach where multiple InPs only map part of the virtual network each. In this scenario, the InPs use a centralised approach, but the overall approach of the solution is still considered distributed.

The advantages and disadvantages of a distributed approach are given in table 2.4 [1,12,22,27].

Table 2.4: Advantages and Disadvantages of a Distributed Approach

Distributed Approach	
Advantage	**Disadvantage**
Distributed algorithms carry out the mapping process in a decentralised way (without a central control entity in charge), and this makes the system more robust	The performance of this approach is usually not comparable to existing centralised approaches, because there is diminished global information
The most significant advantage is the improved scalability because the load is distributed among several nodes and causes the individual nodes to cope with the embeddings effectively.	The most considerable disadvantage is the massive communication overhead. Each node needs adequate information about the global state of the network, since the more information the nodes possesses, the better results will be. The situation becomes a trade-off between communication cost and quality of the embeddings, caused by this increased overhead.
In a variety of resource allocation problems in telecommunication networks distributed resource allocation mechanisms are recognised as providers of low-complexity, efficient and privacy-preserving solutions	These solutions incur a cost though, through the increased coordination messages and increased network overhead
Some argue that since there are a more significant number of coordination messages that are exchanged between embedding control nodes, more information can be collected concerning the global state of the network, which can lead to more optimal embedding.	The increased communication brings the additional load to the SN and this large communication overhead on the SN, as well as the lack of a central entity that has a global view of the system, leads many to believe that this approach invokes non-optimal solutions.

A significant challenge is efficient and effective VNE, some of the difficulties considering the control mechanisms are listed below divided into three main problems [22].

- Resource Evaluation Problem

  To properly evaluate substrate resources is usually a large challenge.

In distributed approaches, there is no central control entity which means that every substrate node must broadcast its status information to others across the substrate network in order to achieve global resource evaluation. This results in a high amount of communication overhead.

Centralised approaches have a global view with up-to-date and accurate status information about each substrate resource. Collecting this information can be difficult with a central control entity for many reasons, such as congestion or delay.

- Information Consistency Problem

  Centralised approaches have a significant weakness concerning the heavy burden of the central control entity. Information is synchronously mapped between substrate nodes to avoid conflicts and inconsistencies that may be present in distributed algorithms; this is done in the aim of alleviating the hot spot problem.

  When the size of the substrate network becomes larger the number of exchange messages grows exponentially, this is the result of a flooding scheme of information synchronisation.

- Mapping Management Problem

  Distributed approaches do not have a central control entity, and without one it is challenging to determine which VN must be mapped first, as well as determining what other management operations must be performed for the VN.

  When substrate nodes are independent and distributed it becomes difficult to change and adjust protocols or policies.

  For this reason, effective policies are considered potentially essential in VNE performance.

Considering all this information, it is not clear which approach is the definite best. Triki et al. [27] propose that the resolution of this dilemma lies in the use of hybrid approaches that rely on clustering techniques. This is done to confine the embedding tasks to a limited subset of substrate nodes and thus reducing the communication overhead and embedding cost.

2.3.3 Static vs Dynamic

VNE approaches can be categorised into another two collections, which is static VNE approaches and dynamic VNE approaches. This division is closely related to the online and offline version of the approach.

Static VNE algorithms contain the assumption that the virtual network demands are known in advance, while dynamic approaches use on-demand mechanisms where virtual network resources are allocated dynamically [23].

Another definition of this classification is that static approaches have consistent, unchangeable infrastructure; and that dynamic approaches consider changes in the virtual or substrate infrastructure. This means that static approaches do not consider the possibility of remapping one or more virtual network requests to improve the performance of the embedding, while dynamic approaches will attempt to reconfigure mapped virtual network requests for the reorganisation of allocated resources and optimisation of substrate resource utilisation [1].

The need for relocation of certain parts of a VN or even an entire VN can be caused by several different factors such as:

- The fragmentation of a SN's resources, since the embedding becomes fragmented over a period. The fragmentation is caused by the arrival and embedding of new VNRs as well as the expiration and release of other VNRs. The fragmentation induces the diminished ratio of accepted VNRs and consequently interrupts long-term revenue.
- The changes in the VN, since the topology, size and resources of a VN can change as time progresses and this is attributed to new requirement demands of users.

- The changes in the SN, since the SN can also possibly be subverted by long-term changes. It is possible for the network infrastructure to be changed or updated occasionally to deal with scalability problems. When network infrastructure is changed (usually by the InP) the SN is changed, and current VNEs will be able to find different or more optimal allocations.

2.3.4 Concise vs Redundant

If the VNE is resilient concerning node or link failures, it is considered redundant. Otherwise, if no redundancy is present the system it is considered concise [1].

The reasoning behind this classification is driven by the fact that the failure of a single substrate entity will affect all the virtual entities that are mapped upon it. In certain environments the deployment of fault sensitive applications considering VNs are essential, and in that environment, it is advisable to procure and develop backup resources that are used as fall-back resources in the case of the corresponding primary resources failure.

A summary of concise approaches is given:

- Only the amount of substrate resources necessary to meet the demand of a VN is used.
- No additional/redundant resources are reserved.
- This means that the resources that would have been saved for redundancy can be used to embed more virtual resources.
- There is no guarantee that a VN will recover from a failure in the case of the failure of some substrate devices.

A summary of redundant approaches is given:

- An approach is considered redundant is it reserves additional resources for the virtual entities.
- These additional resources can be used in case some substrate resources fail at run-time.
- If there is an increase in the degree of reliability of embedding, the embedding cost also increases since the more resources that are used for this purpose; the less virtual entities can be embedded.
- This means there is a trade-off between the reliability of an embedding and its embedding costs in general.
- In this approach, the substrate entities must be monitored once the embedding is completed.
- The driving motivator of this approach is the idea that in the case of failure of an instance, a fallback mechanism must exist that switches the failed primary instance to one of the backup instances and complete its activation.
- For example, once the fallback mechanism activates it must update appropriate routing tables.
- When an embedding algorithm maps virtual links to specifically one communication path in the SN it is called a single-path approach.
- The version of embedding algorithms that map virtual link demands onto multiple paths is also considered to fall in the redundant category.
- These multi-path approaches might split demanded bandwidth of virtual links to multiple substrate paths, thus if some of these paths experience failure packages can still be routed through the remaining communication links that have been set up between the substrate nodes and this is achieved by changing the splitting ratio.

- Multi-path approaches are fully transparent since no reconfigurations need to be completed. Some care must be taken considering bandwidth constraints in order to avoid overloading of substrate link capacity.

This classification pertains to the physical implementation of VNE and is thus not a crucial factor concerning the nature of this dissertation.

2.3.5 Robustness

Many literary works assume that input data is precisely known and this assumption can be difficult to achieve in real life. An example of this is the challenge presented when speculating the required resources of each virtual entity or the power consumption of servers beforehand in a manner which would be considered accurate. For these reasons, problem solutions are sometimes useless in practice because small changes in input data values lead to the infeasibility of found optimal solutions since the presence of uncertain data in these optimisation problems is not considered. Models that take data uncertainty into account must thus be developed, such as Stochastic Programming or Robust Optimisation (RO). Regarding RO, the robustness of a system is attempted against uncertain or deterministic variability in input parameters. RO also contains the assumption that the probability distribution of uncertain data is not knowledge accessible beforehand and the uncertain data, therefore, is assumed to belong to a so-called uncertainty set, which is unlike stochastic optimisation approaches.

RO is thus found by an uncertainty set, and this set identifies the deviations of coefficients against nominal values. Consequentially solutions that are considered robust are by design deterministically immune to realisations of uncertain parameters in certain sets [28].

The degree to which a system can correctly function while experiencing a shift from the expected parameter values is what is defined as robustness. It is considered an important area of research to determine if a resource allocation process is robust against uncertainties since in this scenario the execution time of a tack can vary significantly based on factors like process data content. Since robustness is also defined as the degree to which a system can function and maintain a certain level of performance, even with uncertainties, it is an interesting research topic because heterogeneous, parallel and distributed computing systems frequently operate in environments where it is common to experience uncertainties in task execution time [29].

There are three fundamental questions that a system must answer for that system to be classified as robust. These motions are [29]:

1. What behaviour makes the system robust? The allocation that is capable of task execution by its deadline is considered robust resource allocation in this environment.

2. What are the uncertainties that the system is robust against? This is represented by a random variable. Authors have defined their uncertain variable as execution time for each task on the machine, since it is an acknowledged source of uncertainty.

3. How is system robustness quantified? The robustness can be quantified through the expected number of tasks that will complete before their individual deadlines and can be predicted at any point in time.

Using these three questions one can formulate a resource allocation robustness measure for a system.

The reason why robustness is considered an important factor in some works is that the ability to handle uncertainties is sometimes a crucial factor since their failure can hold dire consequences for people, industries, and public services, such as with mission-critical systems like supply systems. In these specific systems even though there are fluctuations and unexpected events, the demand must be met ensuring the scalability and availability of a system remains a major consideration [16].

2.3.6 Survivability

There are many types of scenarios of failure in network embedding problems that have been investigated, such as single link, single facility node and single regional failure [25]. When the faults of physical network resources are taken into account, the problem is referred to as the Survivable Virtual Network Embedding (SVNE) problem [30] and has recently attracted a lot of interest [25]. SVNE has the intention to assign substrate resources for virtual requests efficiently, as well as to recover mappings on failed or broken substrate resources in the case of a node or link failure [30].

Some past works have treated all physical failures as link failures; this provided backing up of physical resources in advance by focussing on the provision of protection for the substrate network or enhancing the VNs [30].

Some works consider a high recovery ratio as an important goal and use recovery methods that try to recover as many as possible virtual requests through determination of substitute nodes and paths [30].

In works that consider the SVNE problem has the added attribute to be able to accommodate not only the active demands but the backup demands as well. Hu et al. [25] studied a similar problem and took into account node survivability.

2.4 VNE Approach Considerations

2.4.1 Energy Awareness

An emerging field of interest in the VNE field is green networking or energy aware VNE. [1]

Information and communication technology (ICT) is estimated to account for 4% of the worldwide energy consumption. This percentage of energy use is expected to be double by 2020. For this reason, to achieve sustainable ICT, it is important to design and operate converged IT and optical network infrastructures in manners that are power saving. In this specific context, some works have shown huge energy savings when IT processing jobs are allocated in an energy-aware fashion, by using relatively low energy consumption optical networks [31].

The future Internet has a key issue that some works try to address, which is the reduction of energy consumption. It is approximated that in 2011 the Internet was responsible for 2% of the overall energy consumption. Certain works have also stated that the greenhouse gas emissions must be decreased by 2020 by a volume of 15–30%, to ensure the global temperature does not increase more than 2 °C [32].

The authors in [32] concur that many works have started to focus on researching energy consumption reduction technologies of the Internet. The reason for this research field is that Internet resources are frequently supplied for peak or maximum load, and this means these resources are not properly utilised during normal operations, and insufficient and much energy is wasted; this is why current power consumption of network equipment is considered insensitive to traffic load. The increasing electricity costs have become a burden for InPs [33]. Thus it can be said that there is a large room for improvement for energy savings in these specific problems. Intelligent energy-aware network deployment is enabled through resource consolidation. Thus keeping the network performance constant while switching off or hibernating as many network nodes and links as possible is considered one of the best approaches to minimise energy consumption.

InPs manage the shared SN, while SPs create VNs. In VNE VNs that have constraints on both nodes (such as CPU) and links (such as bandwidth) are embedded into the same shared SN, which is why it is considered Energy Efficient Virtual Network Embedding (EEVNE) when the aim is to reduce the energy consumption of the SN when the VNRs are embedded onto the substrate SN [32].

Melo et al. [9] agrees that Network Virtualization is a key component of the Future

Internet. Previous works have either designed heuristic-based algorithms to address the efficient embedding problem or to address the energy impact, this paper aimed to solve the online VNE as an optimisation problem while aiming to minimise the energy consumption and optimise the resource allocation per VN mapping request. This paper used two different objective functions: (i) Bandwidth Consumption Minimization (BCM) - to address resource consumption problem; (ii) Energy Consumption Minimization (ECM) - to address the energy consumption problem. These two objective functions were compared to an already established objective function, the Weighted Shortest Distance Path (WSDP) function, that was considered a state of the art solution in the resource allocation field. The reduction in energy consumption, embedding factor, resource consumption and acceptance ratio were all evaluated, and it was shown that these proposed functions improved on the current established solution.

2.4.2 Security and Protection Levels

Another emerging field of research considering VNE pertains to security. Virtual networks can be hosted on the hardware of multiple different InPs. At the same time, one InP can host networks from multiple Virtual Network Provider (VNP)s/Virtual Network Operator (VNO)s, since VNOs are foreseen to rent infrastructure via VNPs from InPs in a virtualized future Internet. This setup means that both parties - the VNOs and the InPs - posses an interest to protect their respective assets (i.e. nodes and links – either physical or virtual) [1]. VNOs can distrust each other and thus insist that their virtual infrastructure is not co-hosted on the same physical equipment as other operators, and this creates an open research issue [2].

The network architecture is injected with an additional virtualization layer and multiple VNs cohosts on shared network infrastructure. Network virtualization creates a new range of security vulnerabilities that are not present in current antiquated network architectures. These risks can be divided into three categories [20]:

1. Physical hosts attacks on their VMs

In the SN the physical hosts are responsible for offering resources to VMs in the VN, this is completed within a certain service level agreement. Thus ultimately the physical hosts carry out all the services and applications operated on the VMs. When malicious attackers control a physical host, it can modify the aspects of the VMs hosted on it as well as monitor traffic associated with the hosted VM, modifying legitimate traffic or injecting malicious traffic that can disrupt the functionality of the VMs. Since these VMs are supervised by their hosts all the time, the VMs are not capable of defending themselves against attacks from their physical hosts.

2. VMs attacks on their physical hosts.
   When a VM is malicious, it can access the vulnerabilities of its host through the allocated resources. This can break the fixed isolation enabled by the virtualization process. Malicious VMs can attack the network infrastructure through intrusion or taking control of the physical host. These attacks disrupt services hosted by competing VNs, such as the VM can launch a denial-of-service (DoS) attack against the physical hosts, which will remove all the available resources from the hosts, and eventually bring down the entire network infrastructure

3. VMs attacks on other VMs that coexist on the same physical host.
   VNs are logically isolated from each other in the network virtualization environment, while VM nodes located in different VNs can share some of the same resources of the physical hosts. Malicious VMs can take advantage of shared resources to gain admission to cohosted VM vulnerabilities; they also possess the ability to launch a cross-VM side channel attack to steal information from vulnerable VMs that coexist on the same physical host

In virtualized environments, there are traditional protection mechanisms that can be helpful, but these carry no guarantee to be successful and can obtain a high-security overhead [20].

Thus in this environment, different stakeholders each bear their security concerns. A VNO may want their network hosted on hardware that can offer a sufficient level of security, but a InP would be interested in ensuring that VNs are properly secured and do not run havoc on their equipment [1]. Some VNOs may distrust each other and require that their virtual infrastructure is not co-hosted on the same physical equipment to minimise the risk of cross-virtualization attacks [19].

These concerns cannot be solved simply through the installation of additional software on either the virtual or physical nodes; rather it becomes essential to avoid mappings that will increase the risk for one of the stakeholders. This is the reason that security aware VNE algorithms are an open research field. An algorithm is considered a security-aware VNE algorithm if it considers there requirements while trying to minimise the risk exposure for all involved stakeholders. These research works state that is it essential for VNOs and InP to be allowed to express their security needs, and VNE algorithms must aim to match those requirements as closely as possible [1].

2.4.3 VNE Performance Metrics

VNE is a critical component of future internet structures and provides end-to-end services to users while achieving optimisation goals. The optimality of the solutions must be measured, and this is achieved regarding various performance measures, such as QoS, Quality of Network Economics (QoNE), Quality of Resilience (QoR), security, energy efficiency, acceptance ratio, throughput, algorithm runtime and other metrics [23]. Performance metrics are needed to enable the evaluation of the quality of successful embedding and to compare different VNE approaches in a standardised manner.

The performance metrics summarisation is given in table 2.5, table 2.6, table 2.7 and table 2.8, and are a compilation of various research works [1,3,13,20,21,34]

Table 2.5: VNE Performance Metrics: Quality of Service QoS)

Quality of Service Metrics (QoS)		
Metric	**Definition**	**Comments**
Path Length	Measures the number of links between two substrate nodes that are mapped to two interconnected virtual nodes	Longer path length means more resources must be reserved for virtual link embedding
Stress level	Number of virtual entities that are mapped onto that substrate entity	The higher amount of virtual entities that use the same resources, the larger impact when considering possible side effects
Utilization	the sum of spent substrate resources (due to the mapping of virtual entities) divided by the total amount of resources, for each SN entity (node or link)	Considers the magnitude of the resource usage, not just the number of virtual entities that use a resource, making it a more precise measurement than the stress level
Throughput	Maximal data rate that can be transmitted through all connections between source-destination pairs.	Measured after embedding of VN has been performed
Delay	The amount of time needed for a packet to travel from one node to another in the network.	VNE algorithms that be created to minimise the delay between virtual nodes when optimising the mapping of virtual resources
Jitter	Variance in packet interarrival times	Can be integral to the substrate network (e.g. due to weak links) or introduced by virtualization itself (e.g. due to concurrent resource usage by two different virtual entities). So far no VNE approach focuses on explicitly minimising jitter (to the author's knowledge).
Security	Operators must ensure that the security features of their network will not be affected when deploying VNFs.	Virtual entities can run in data centres that are not directly owned by the NO and can also be outsourced to third parties, and this is the reason why NFV can contribute to new security concerns, even though NFV creates many benefits.

Table 2.6: VNE Performance Metrics: Quality of Network Economics (QoNE)

Quality of Network Economics (QoNE) /Cost-related Metrics		
Metric	**Definition**	**Comments**
Cost	Amount of substrate resources used for VNE	Usually determined by summing up all CPU and bandwidth resources of the substrate network that have been reserved for VNRs. Besides these two resources, other types of resources could also be considered. Likewise, different types of resources can optionally be weighted in dependence on their value range. It is directly related to the length of substrate paths, since the longer a path, the more substrate resources are needed and the cost of the embedding increases.
Revenue	Sum of virtual resources that were requested by the virtual entities in reality	Usually computed using the same strategy as determining the cost
Cost/Revenue	Revenue is typically taken into account in addition to costs. By dividing the cost by revenue, varying cost values are balanced.	The higher the value/ratio, the more resources were needed to embed the VNs. This ratio is used to compare different embedding algorithms in a meaningful way.
Acceptance ratio	Number of virtual network requests (that could be embedded) divided by the total number of virtual network requests	Used to evaluate the ratio of the VN requests that are successfully mapped
Resource Utilization	Long-term revenue to cost ratio	The revenue of accepting a VN request at a specific time is defined as the total resources it requires. High-security demand cause high revenue, this is why long-term average revenue is defined as resource utilisation.

Table 2.7: VNE Performance Metrics: Quality of Resilience (QoR)

Quality of Resilience (QoR) Metrics		
Metric	**Definition**	**Comments**
Number of backups	Counts the number of backup resources that are set up for a virtual entity	In the scenario of the entity hosting the virtual entity failing, various additional substrate entities can be reserved to act as a replacement
Path redundancy	Ratio between the number of backup paths to the number of direct paths	Refers to the number of additional resources used to back up the embedded network and specific redundancy algorithms establish backup paths to be used in case some parts of the network break down
Cost of resilience	Measures the ratio of total number of running nodes and number of backup nodes	Related to some additional nodes required to maintain resilience. Does not focus on connectivity resources, contrary to path redundancy, but does include demand node resources
Recovery blocking probability	Ratio of the number of unrecoverable failure scenarios to the total number of failure scenarios	Re-organization occurs when a substrate entity fails and the SN must recover from failure, mainly compensatory resources must be allocated.If an approach does not reserve extra capacities in advance, the system must identify suitable backup resources at runtime, since the entities of a system are limited in their capacity this approach might fail, and thus results in a failure of recovery.
Number of migrations	In the case of substrate node failures, it is the number of virtual nodes required to migrate to new facility nodes	When a substrate node fails, the virtual node hosted on it must at least be moved. Other constraints can trigger even more migrations, such as maximum path length. Migrations are resource intensive. Thus they should be kept to a minimum

Table 2.8: VNE Performance Metrics: Other

Other Metrics		
Metric	**Definition**	**Comments**
Runtime of the algorithm	Compares algorithms concerning the time they need to compute an actual embedding result	Runtime is a crucial factor in most real-life systems and especially in data centres. VNE is considered an NP-hard problem and suffers from intractable computation complexity with a large search domain. Even though by sacrificing the optimality of results, some researchers have proposed many heuristic algorithms to improve computation efficiency, the computation is still not efficient enough, and the search space is still too large.
Number of coordination messages	Used in distributed embedding approaches. Comprises of the different messages that must be exchanged between substrate nodes for coordination purposes	Can be used to determine and compare communication overhead between different distributed approaches
Active substrate nodes	Related to the average length of substrate paths	Since additional nodes are used to forward communication data between end nodes, the possibility arises that previously switched off nodes could be selected to forward data.
Manageability	A VN should be instantiated at the right time and in the right location, have hardware resources dynamically allocated and scaled, and be interconnected in order to achieve service chaining.	Management of virtual and legacy appliances are presented with some new requirements due to this flexibility of service provisioning. In data centres hardware resources are almost equivalent, and this produces more natural coordination, this is not true in NFV and is the reason why manageability in NFV is somewhat dissimilar to data centre management. The reason for the difference is that the cost and value of resources can vary significantly between network points of presence and customers' premises. These various fluctuations must be considered by the management functionality in order to optimise resource usage across a wide area.

2.4.4 Modelling of VNE Problem

VNE algorithms aim to achieve certain objectives, as discussed in previous sections. These objectives can be divided into certain categories such as QoS, QoNE, QoR as well as other measurement tools. Newer research works focus on some new objectives such as energy awareness or improved security for the optimisation goals [27]. Depending on the situation objectives are chosen for a VNE algorithm, and this influences the manner in which a model is designed.

The VNE problem has been divided into two phases that were considered organic; these were the node mapping phase (also referred to as the VNoM problem) and the link mapping phase (also referred to as the VLiM problem). In this node mapping phase, CPU requirements are satisfied while selecting substrate nodes for virtual nodes. In the link mapping phase, bandwidth requirements are satisfied while virtual links are mapped onto physical paths that are between selected substrate nodes [24].

In previous solution approaches to the VNE problem, there have been some main embedding objectives. Although some of these only focussed on solving the VNoM and VLiM problems, which does not consider solving the VNE problem in a coordinated fashion. These main embedding objectives were [1]:

- Providing QoS-compliant embedding;
- Maximising QoNE metrics for the gain of the InP, and
- Providing SVNE.

There are some restrictions on the problem space that have been implemented to reduce the hardness of the VNE problem and to enable efficient heuristics. These restrictions include [7]:

- Assuming all the VNRs are known beforehand, thus considering the offline version of the problem.;

- Disregarding either node or link requirements;
- Eliminating admission control through the assumption that the substrate nodes and links have an infinite capacity, or
- Only focussing on a certain topology of the VN.

All of these approaches aim to solve the VNE problem efficiently and effectively. These approaches change how the model designed, and the solution is created. Previous works completed in this field have modelled this problem in different manners concerning certain sections of the VNE problem. Some of these modelling approaches that researchers have used in summarised in table 2.9.

Table 2.9: Modelling of the VNE Problem

VNE Modelling	
Section	**Approach**
Virtual Network	topologies randomly generated [13, 20, 27] ; number of virtual nodes follow a uniform distribution between 2 and 10 [13] ; each pair of virtual nodes are randomly connected with probability 0.5 [13] ; contains four nodes and five links [27]; contains maximum of 7 nodes and 12 links [35].
Substrate Network	randomly generated topology [20]; topology of 100-nodes transit-stub topology generated by the GT-ITM tool [27]; Contains 100 nodes and about 500 links (about the size of a medium-sized ISP) [20];
CPU capacity	resource of substrate nodes and is constraint each node has [20, 27]; are real numbers following a uniform distribution between 50 and 100 [20]; uniformly distributed between 0 and 30 [13]; random value in the range of 100–200 GHz [27];
Bandwidth Capacity	resource of substrate links [20] are real numbers following a uniform distribution between 50 and 100 [20]; between 0 and 50 [13]; generated randomly between 500 and 1000 Mb/s [27]
Links	virtual links associated with bandwidth capacity, permissible delay and packet loss constraints [27]; each substrate link delay and packet loss tolerance are random values in ranges of 100–350 ms and 0% and 5% respectivelyTriki.2015;
Implementation environment	Matlab [27]; CPLEX [35]
Virtual Network Requests	subject to 50 requests [27]; arrival of requests follow Poisson process (average arrival rate of 5 per 100 time units) [20]; lifetime of request modeled by exponential distribution (average of 1000 time units) [20]; arrival rate (number of VN requests coming per window period) is a Poisson process varying from 4 to 10 requests per window [13];

2.4.5 Computational Complexity

Chowdhury et al. [7] put forward that theoretical upper or lower bounds do not exist for the general version of this problem. If some constraints are ignored, the VNE problem is still defined as NP-hard. For this reason, the problem space in heuristic algorithms was restricted in early research on this topic. This restriction only considered link constraints and adopted the view that virtual nodes were assigned to the SN [24].

The VNE problem has been proved NP-hard by [19]:

- the reduction from the multi-way separator problem, and
- the reduction from the unsplittable flow problem.

The VNE problem has been proven NP-hard, and as a solution, two different approaches have been proposed. Firstly, to apply techniques such as relaxation and rounding (which is considered a standard technique) and secondly, an approach based on heuristics [36].

Researchers have formulated many different heuristic algorithms to discover practical solutions to this problem since even restricting the problem space does not change the fact that the VNE problem is NP-hard. The main driving motivator behind this research is the intractability of computation [22].

The VNE problem is still considered NP-hard when an offline version of the problem is used. In the unsplittable flow scenario after the completion of mapping all the virtual nodes, the embedding of virtual links that consist of bandwidth constraints onto an underlying physical link or path is still considered NP-hard [7, 19].

2.4.6 Financial

Since there is an absence of a real-world marketplace, it is nontrivial to produce a pricing model that can capture the interactions between buyers, sellers, and brokers in a network virtualization environment [7]. Although this still is true to a point many works have made some progress in creating pricing models for this type of technology.

A InP generates revenue by renting out sliceable virtual resources in order to fulfil a SP's resource requirements. In this structure, the InPs costs are closely related to the used resources. Frequently high acceptance ratios lead to high revenue. However the pricing strategy can also cause significant deviations. Benefits can be maximised at relatively low acceptance ratios with utility functions that have high quality and expensive services [22].

Certain works have made some assumptions considering the pricing structure of the VNE problem. These assumptions include:

- Using the average bandwidth cost to evaluate the embedding cost [13].
- Assuming the cost to install and operate one content delivery network node is set in a range, such as 8 - 12 kUSD, while different prices are considered for virtual content delivery network nodes in the range 0.001 - 10 USD per Mbit/s [37].
- Assuming revenue and cost are linear functions of requested and allocated resources [7].
- Using revenue and cost functions that are independent of VN lifetime [7].
- Considering the lifetime of a VN request as a factor in the way it is priced [7]. However the authors in [7] did not create revenue and cost functions that are independent of a VN lifetime.
- Using a pricing model that prioritises some crucial resources over others and must include resource prioritisation while making an embedding decision [7].
- Accepting that there can exist various pricing models, and creating VNE algorithms that are optimised for each one individually. This can be achieved by allowing an algorithm to accept the updated cost and revenue functions [7].

Zhang et al. [38] aimed to design a novel online stochastic auction mechanism for the NFV market. This was completed in order to provide, and price service chains as well as actively buy additional resources.

2.5 Previous Research

This section provides other information on VNE research.

Previous research focusses on designing heuristic-based solutions to address the practical embedding problem or otherwise to address the energy impact [9].

In order to satisfy different latency requirements, virtual network operators specify different priority classes, these networks may also require substrate resource sharing among these priority classes in order to minimise the number of substrate resources assigned to them [39]. Ogino et al. [39] formulated a novel virtual embedding problem and proposed a heuristic VNE method to minimise the total substrate resources that are needed when operators require physical resource sharing among the different priority classes within their virtual network. This paper proposed that using this method multiple virtual networks could maximally share substrate resources while still sharing these resources on a reasonable basis, but this was dependant on the number of node degrees. The more virtual embedded networks there were, the more effective this procedure became regarding the number of physical resources needed.

2.6 Supplier

A traditional ISP is decoupled into two independent management entities (also called tiers) in the current network virtualization environment; this is done to provide dedicated networks and services respectively [25]. The decoupled entities are referred to as the InP and the SP. This division is similar to the typical 2-layer IaaS business model in Cloud Computing [22].

A 4-layer IaaS model is suggested by many literary works, and this division of the traditional ISP allows for the easy deployment of customised network protocols without the need for universal agreements between competing infrastructure holders. The layered architecture also provides great flexibility and diversity regarding service provision over the Internet [18]. The four layers consist of the acInP, the SP, the VNP and the VNO and figure 2.9 illustrates the current division of an ISP as per this 4-layer IaaS model definition [1].

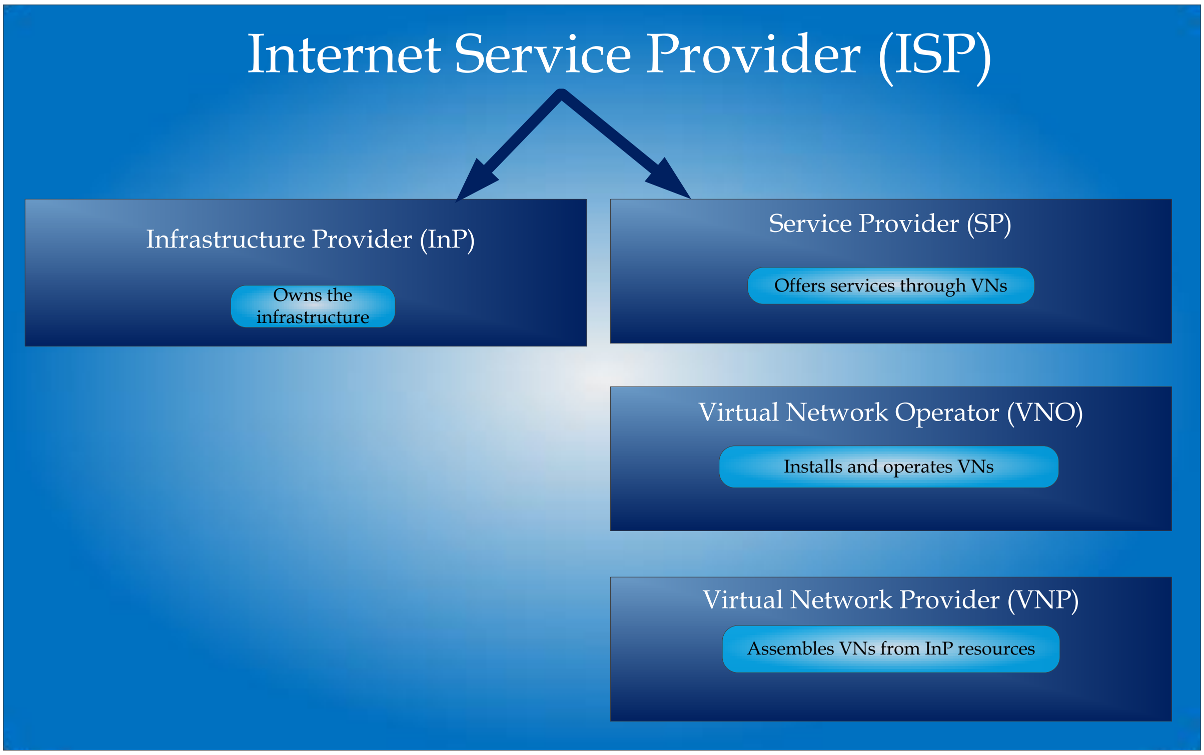

Figure 2.9: Illustration of ISP Framework

The divisions of the traditional ISP in this model are listed and discussed.

Infrastructure Provider InP [4,7,18,19,22,25]:

- Physically deploys the network equipment;

- Manages the network equipment ;
- Maintains the network equipment;
- The network equipment are physical resources that form the substrate network, such as physical server nodes and communication links;
- Has significant invest;ment in physical infrastructure since they own physical resources such as network equipment and transmission lines
- Serves different SPs, and
- Through the separation of services from the supporting networks physical network resources can be abstracted by the InP to virtual network resources, which can then be offered to SPs for the objective of being shared as independent slices, such as multiple VNs.

Service Provider SP [1,4,7,18,19,22,25]:

- The SP is free of management and can concentrate on the business of offering customised services;
- Creates and offers services through virtual networks;
- Responsible for the deployment of network protocols;
- Offer end-to-end services, building on aggregated resources ;
- Establishes agreements with one or more InPs;
- Uses InPs for access to a subset of physical resources, and
- Purchases slices of substrate resources (e.g., CPU, bandwidth, and memory) from the InP and then create their customized VN to offer value-added service (e.g., Voice over IP, online video, online gaming, content distribution and other Over-The-Top services) to end users without the need to pay attention to the details of the SN.

Virtual Network Provider VNP [1, 4, 19]:

- Assembles virtual resources;
- Collects and manages physical resources, and
- This is done from one or more InPs.

Virtual Network Operator VNO [1, 4, 19]:

- Makes use of underlying physical resources to create virtual networks, and
- Installs manages and operates the virtual network according to the needs of the SP

Some work consider another division of this model, by decoupling the InP into two separate entities [27], namely:

- The Physical Infrastructure Provider (PInP) who would own and manage the substrate resources as well as have the ability to virtualize and partition these substrate resources into slices that may be allocated to VNs.
- The Virtual Infrastructure Provider (VIP) who would represent the main role in VN environments by discovering and aggregating virtual resources from one or more Physical Infrastructure Providers. This is done with the aim of instantiating VNs that would satisfy the SPs demands.

The SP in this model is not subdivided, such as the 4-layer IaaS model mentioned earlier, but this decoupling is not widely considered normalised.

2.7 Chapter Summary

The field of VNE research has many components. In this chapter, the VNE problem is discussed through an introduction to its enabling technology: network virtualization.

Several VNE features are considered such as the VNE problem, resources and parameters in the VNE problem and problem decomposition and coordination. Different VNE classifications are presented such as online vs offline, centralised vs distributed, static vs dynamic, concise vs redundant, robustness and survivability. Other VNE approach considerations are covered as well, such as energy aware and secure VNE, VNE performance metrics, computational complexity and financial features of VNE. Previous research is discussed throughout this chapter, and the supplier in this specific problem is addressed.

Chapter 3

Literature Study - Optimization Model

In this chapter optimisation techniques that are commonly used are discussed. In this study, linear programming and its sub-techniques are explored as well as solution methodologies and modelling techniques that can be used in this dissertation. Stochastic programming and resource allocation are also explored.

3.1 Computational Complexity

Theory of computation is a section contained within theoretical computer science and mathematics, and a subsection of this is referred to as computational complexity theory. This section concentrates on classifying computational problems according to their difficulty, as well as how these problem classes relate to one another.

The mechanism of mathematical step application is defined to be a solving catalyst for computational problems, such as an algorithm. This environment creates the possibility for Non-Deterministic Polynomial-Time (NP) problems to be defined. The reasoning behind the naming of this term is the original application to non-deterministic machines. However, NP has been classified as an essential complexity class considering computational complexity theory.

The NP problems can be considered a set of all decision problems, in which each instance that provides a positive result will have efficient proofs that can be verified and thus fully confirming the answer. In other words, a deterministic Turning machine should be able to verify these proofs in polynomial time when considering this problem from a mathematical point of view. The computational complexity can be divided into two categories, namely NP-hard and NP-complete [40].

3.2 NP-hard

"Assume there exists a problem H. H is considered NP-hard if and only if every problem L in NP can be transformed in polynomial time to H." This is how NP-hard is defined in computational complexity theory [40]. This means that if there exists an algorithm to solve a problem A, and that algorithm can be translated into one that can be used to solve any NP problem, that problem A would then be classified as an NP-hard problem. NP-hard may, therefore, be defined to be "at least as hard as any NP-problem".

3.3 NP-complete

"If a decision problem is found in both NP and NP-hard, it is classified as NP-complete." This is how NP-complete is defined in computational complexity theory [40].

An acknowledged component of the NP-complete problem is that there exists "no fast solution" for them. These problems can verify a solution in polynomial time, yet there exists no efficient method to obtain a first off solution. An essential factor to consider, as well, is that as the size of the problem increases, the time required to solve that problem also increases proportionally.

3.4 Linear Programming

A Linear Programme consists of continuous variables and linear constraints and these can be inequalities or equalities respectively. Optimising the linear cost function is the objective of this formulation. The purpose of the objective function is to either achieve a maximum or minimum cost. One of the standard forms of LP is given as [41]:

$$\begin{array}{rllll} min & c_1x_1 & +c_2x_2 & +\cdots+c_nx_n & \\ subject\ to & a_{11}x_1 & +a_{12}x_2 & +\cdots+a_{1n}x_n & =b_1 \\ & a_{21}x_1 & +a_{22}x_2 & +\cdots+a_{2n}x_n & =b_2 \\ & & \vdots & \vdots & \\ & a_{m1}x_1 & +a_{m2}x_2 & +\cdots+a_{mn}x_n & =b_m \\ & x1,\cdots,xn & \geq 0 & & \end{array}$$

And this form can be reduced a matrix notation:

$$min\{c^T|Ax=b,x\geq 0\} \tag{3.1}$$

where $c, x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ and $A \in \mathbb{R}^{m\times n}$. In this notation the c refers to the cost vector, while the vector of variables is represented by x.

The standard format of a LP formulation consists of two elements: Firstly, the objective function, which is subject to the second element, the constraints. The objective functions state what the formulation aims to achieve and the constraints form the boundaries of the model space and using these two elements it is possible to obtain a feasible region where the optimal solution can be found. This concept is demonstrated in figure 3.1 [41].

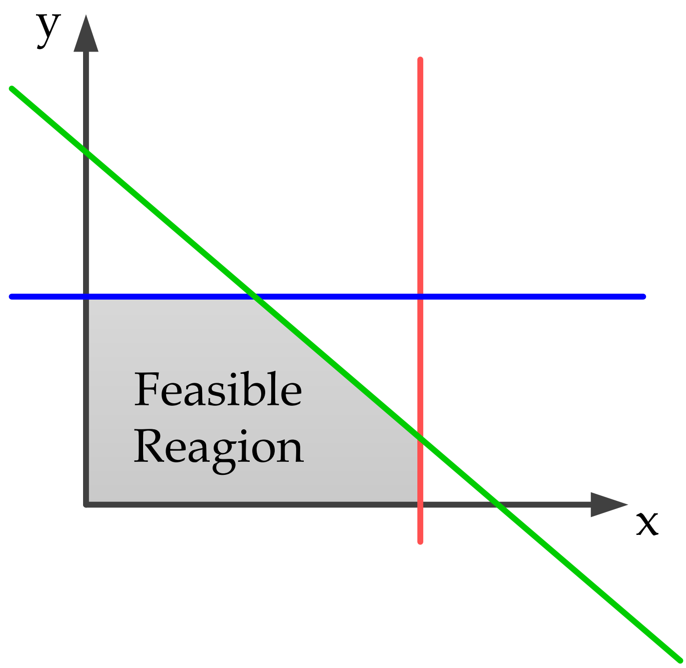

Figure 3.1: Identifying the Feasible Region in Linear Programming

The objective function and constraints that the LP formulation comprises of both consist of unknown variables. The problem is referred to as a Integer Linear Programming (ILP) when the model requires all the variables be of an integer type. The problem is referred to as Mixed Integer Linear Programming (MILP) if only a few of the variables are required to be of integer type. ILP and MILP are considered NP-hard in various practical situations; this is in contrast to the LP formulations, in which worst-case problems can be efficiently solved.

3.5 Solution Approaches

VNE is considered a multi-objective optimization problem [22]. Various embedding methods and algorithms can be grouped into four categories which is dependant on the nature of the method itself. These four categorisations are as follow [34]:

1. Heuristic
   In these approaches, the VNE problem is solved using simple algorithms, which is why this approach performs the VNE quite fast and possibly sub-optimal embedding solution is obtained, e.x. the G-SP method.

2. Heuristic combined with MIP
   In this approach, the VNE is performed in two steps. Firstly, a mathematical model is created and utilised to map virtual nodes onto physical nodes, and secondly, MIP is executed to embed virtual links, e.x. the G-MCF method.

3. Heuristic combined with MIP with added improved coordination between mapping phases
   In this approach, the same concept is used as the previous category, but using an augmented "substrate graph construction" improved coordination between mapping phases is achieved, e.x. the R-ViNE, D-ViNE, D-ViNE-SP and D-ViNE-LB methods.

4. ILP
   In the integer linear programming approach, the VNE problem is solved through the use of ILP. Through the combination of a load balancing strategy and minimisation of resource consumption, this approach obtains an optimal solution for a given cost function.

Fischer et al. [1] states that the solution approaches must be classified into the following categories:

1. Exact solutions
   These solution approaches suggest optimal techniques in order to solve small instances of the problem as well as to create baseline solutions that represent an optimal bound for heuristic-based VNE solutions. LP can be used to achieve optimal VNE solutions, and specifically, ILP can be used to formulate the VNE in a coordinated manner optimally. ILPs can be NP-complete in many practical

environments. However, some exact algorithms can solve small instances of the problem in a sensible amount of time, such as branch and bound, branch and cut and branch and price. The solvers used to implement these algorithms (software tools) can be open-source or proprietary. CPLEX [42] is the software tool that will be used in this dissertation. Examples of exact solutions are cutting plane and branch and bound.

2. Heuristic solutions
   A significant factor in VNE is execution time since network virtualization considers dynamic online environments where the arrival time of a VNR is not known ahead of time. Thus the execution time of the VNE should be minimised to avoid a delay in the embedding of new VNRs. For this reason, heuristic solutions have proposed that aim to find an acceptable solution through compromising the solution optimality in exchange for shorter execution time.

3. Metaheuristic solutions
   Since VNE can be considered a combinatorial optimisation problem in which an optimal solution is sought over a discrete search-space, the optimal solution for large instances of these problems can be hard to calculate. Metaheuristics are proposed and may be used to achieve near-optimal solutions; this is achieved through the improvement of a candidate solution with regards to a given measure of quality. Examples of metaheuristics include simulated annealing, genetic algorithms, ant colony optimisation, particle swarm optimisation or tabu search.

As is seen, there are different categorisations for VNE solution approaches which have their respective benefits and disadvantages. These solution approaches were developed over time, as can be seen from the evolution from exact and straightforward methods to more modern, sophisticated methods.

3.6 Optimization Theory

There are many fields in which optimisation theory can be a vital role player, such as chemical engineering, fleet management and inventory organisation since optimisation theory leads to the best possible solutions for a given problem.

Optimisation theory can be a powerful tool for the development of optimal methods and has been successfully utilised in many areas of communication networks, such as optimal routing, flow control and power control.

Initial works based on optimisation techniques on link capacity assignment, routing and flow control in communication networks can be found in many older works.

An application of optimisation theory is the application to resource allocation, in which this technique ensures that resources will follow their QoS requirements and achieve a notion of equity. This is obtained through the appropriate formulation of the optimisation problem, as well as the use of specific allocation policies by the desired type of equity.

As the number of applications competing for resources increases in the current communication network, there arises a need for efficient resource allocation mechanisms to maximise user satisfaction; and optimisation theory can facilitate the tools needed to develop these mechanisms that must allocate resources of a network optimally and fairly amongst its users.

Optimisation theory has been found comprehensively useful in the RAP field of research. This is because optimisation theory can lead to the creation of distributed algorithms that can assure optimal allocation of resources in a network [10].

3.7 MILP

Integer Programming (IP) and MIP approaches have been applied to various RAP and optimisation problems in the area of networking [7].

Solving a MILP is considered NP-hard. The exact algorithms discussed earlier in this chapter have the potential to obtain optimal results. However, they can incur exponentially increasing running time. For this reason, these approaches cannot scale to be implemented on massive VNE problems [20].

A problem can be defined as a MILP problem if there are only some variables that are required to be of an integer type and using this approach ensures sudden changes can be implemented without issue. There are different reasons why MILP as an automation tool is considered a suitable choice in situations such as these.

3.8 Stochastic Optimization

In complex systems - such as communication networks and unmanned systems - there exists an extensive class of combinatorial optimisation problems that are referred to as SRA problems. In this problem environment, there is a certain probability prescribed to each resource's ability to complete a task, and the aim is to maximise the payoff by suitably assigning available resources to different tasks [6].

Anastasopoulos et al. [31] note that "the design objective of stochastic VN planning is to identify the topology and determine the virtual resources required to implement dynamically reconfigurable and dependable VNs over the underlying converged optical network and IT resources." While the user needs must be met for each VN, resource utilisation must be supported efficiently as well.

When VNs are mapped and planned ahead of time for the highest number of requests that have to be hosted, the hosting is determined independently of scenarios. This environment is referred to as static planning.

In another context is deterministic planning, which is the delusional case of idealism, where accurate information of traffic requests is known beforehand and is available to the system.

From these definitions, it can be deduced that optical network resources that are assigned to VNs are significantly lower in stochastic planning than when using static planning. It also observed that the performance in stochastic planning is similar to the ideal deterministic planning case where demands are known accurately in advance. This will, in turn, achieve near-optimal resource efficiency. Furthermore, the benefits of stochastic planning can be developed in practice through the use of dynamic and periodic replanning of the VNs over the SN [31].

It is commonly known that there exists uncertainty in real-world practical situations and for this reason, the stochastic optimisation (programming) is favoured for most SRA in many situations. There are two categorisations of SRA [6]:

- Deterministic SRA (DSRA), and
- Stochastic SRA (SSRA).

Usually, SRA in the context of complex systems receives uncertainty from different sources; this is the reason why SRA is in most cases a stochastic optimisation problem with multiple uncertain parameters. These kinds of stochastic SRA problems are addressed with the use of sampling-based simulation, and this yields different combinations of uncertain parameters which leads to various scenarios; each scenario represents one possibility of the uncertain SRA problem [6].

Mangili et al. [37] undertook the stochastic planning problem specifically for content delivery to study potential benefits thatNFV can achieve for content distribution purposes.

Since planning is performed on a long-term basis, the theoretical framework of stochastic optimisation can be used to guarantee solution robustness, concerning the uncertainty in the probabilistic description of future traffic demands; and some authors use this approach in their work. In real-world environments, future traffic demands used for network design is not usually known beforehand, but some advanced optimisation techniques must consider the stochastic nature of input parameters [37].

3.9 Chapter Summary

In this chapter, the literature regarding optimisation theory and stochastic techniques were discussed. The computation complexity environment was discussed as well as solution approaches to these classes of problems.

Chapter 4

Model Formulation

The model formulation is completed in this chapter. The performance measures that will be used to compare results of this experiment will also be discussed.

4.1 Multi Commodity Flow Formulation

The MCF problem is used to maximise the sum of the flow values of the individual commodities in the network. When considering a LP, the MCF problem can be solved in polynomial time, which is advantageous.

In the MCF problem, each commodity (s_i, t_i, d_i) is composed of:

- a source node s_i;
- a destination node t_i, and
- a demand d_i.

The objective of the MCF problem is to minimize the cost of routing a set of commodities simultaneously in the network, subjected to capacity constraints [43].

4.1.1 Model Verification and Validation

Verification refers to checking or affirming that a proposed solution to a problem is correctly implemented.

Validation implies checking or affirming that the proposed solution to the problem solves the problem or is a valid solution.

For analytical approaches, verification will refer to solving the equation correctly, and validation will refer to whether the correct equation is being solved. Two case studies will be used to verify and validate the model.

Case study A is the first case study and is a small and simplistic network with few requests and only two scenarios. This small case study is divided into two subcategories: a worst-case analysis and stochastic analysis. In the worst-case analysis, there will be no stochastic elements used. Thus no scenarios will be used for the network request. This simple case study is used to verify the model since the outcome of the resource allocation can be calculated easily by hand as well. Thus the effectiveness of the model can be tested on an elementary scale.

Case Study B is the second case study and is a more extensive network than case study A, thus more nodes and links will be present. This case study will also comprise more request and more scenarios. The more elaborate nature of this case study is used to test the validity of the model since the scalability of the model will be tested. This case study is also divided into two subcategories: a worst-case analysis and stochastic analysis, with the same features as in case study A.

4.2 Mathematical model

Let the substrate infrastructure be represented by $G^s = (V^s, E^s)$, with node capacities m_i for all $i \in V^s$, and link capacities n_{ij} for all $(i,j) \in E^s$. Given a virtual network topology $G = (V, E)$, edge $(u, v) \in E$ has a capacity C_{uv}.

Let $R = \{1, ..., |R|\}$ be the set of VNRs. Each request $r \in R$ has a set of virtual nodes $V(r)$.

Stochastic scenarios are defined as the set $S = \{1, 2, ..., |S|\}$. Each of these scenarios consists of different demands for the same request. An example of this would be the difference in demand for a virtual network of business during and after lunch hours.

There is a profit attached to each VNR given as p_{rs}, for each request $r \in R$ and its scenarios $s \in S$. This profit is calculated as the cost of hosting the VN subtracted from the earnings gained from the user of that specific VN. The costs of hosting a virtual network is seen as the cost of using nodes and links for a specific goal.

Set $K = \{1, 2, .., |K|\}$ represents a collection of commodities between two nodes and has a demand defined by d_k^{rs}, for commodity $k \in K$, request $r \in R$ and scenario $s \in S$.

Let the decision variable $y^r \in \{0, 1\}$ receive the value of one if the request $r \in R$ is accepted or null otherwise. Let $x_{ui}^r \in \{0, 1\}$ be one if the virtual node $u \in V$ is placed on the substrate node $i \in V^s$ or otherwise null. Let $f_{ij}^{uv,rs} \in \{0, 1\}$ equal one if the flow of the virtual link $(u, v) \in E$ is placed on the substrate physical link $(i, j) \in E^s$ for a request $r \in R$ and scenario $s \in S$, otherwise null ensuring a link capacity is not exceeded.

The VNE problem can then be represented by the following MILP model:

$$\text{maximise} \sum_{s \in S} \sum_{r \in R} p_{rs} y^r \tag{4.1}$$

subject to the constraints

$$\sum_{i \in V^s} x_{ui}^r = y^r \qquad \forall r \in R, u \in V \tag{4.2}$$

$$\sum_{r \in R} \sum_{k \in K} \sum_{u \in V} d_k^{rs} x_{ui}^r \leq m_i \qquad \forall i \in V^s, s \in S \tag{4.3}$$

$$\sum_{r \in R} \sum_{k \in K} f_{ij}^{uv,rs} d_k^{rs} \leq C_{uv} \qquad \forall (i,j) \in E^s, (u,v) \in E, s \in S \tag{4.4}$$

$$\sum_{(i,j) \in E^s} f_{ij}^{uv,rs} = x_{ui}^r \qquad \forall r \in R, u \in V, (u,v) \in E, s \in S \tag{4.5}$$

$$\sum_{(j,i) \in E^s} f_{ij}^{uv,rs} = x_{vj}^r \qquad \forall r \in R, v \in V, (u,v) \in E, s \in S \tag{4.6}$$

$$\sum_{(i,j) \in E^s} f_{ij}^{uv,rs} - \sum_{(j,i) \in E^s} f_{ij}^{uv,rs} = 0 \qquad \forall r \in R, (u,v) \in E, s \in S \tag{4.7}$$

The objective function (4.1) needs to be maximised to obtain the highest possible profit. This means that the primary goal is to place requests that will earn the most income for the supplier.

Constraint (4.2) ensures that every virtual node is only placed on a single physical node, which safeguards against the possibility of over- provisioning for the virtual nodes. This ensures that only one substrate node will be hosting the virtual node and resources are not wasted.

Constraint (4.3) and (4.4) are responsible for the node and link constraints separately, to ensure that the virtual node and links placed do not exceed the capacity of the substrate infrastructure.

Constraints (4.5) , (4.6) and (4.7) are the flow constraints of the model, these constraints ensure that the flow of the virtual network is achievable on the substrate infrastructure and does not interfere with the other networks that may be present in the system.

Constraint (4.5) ensures the flow at the source node is correct and that a flow must exit its source node completely. Constraint (4.6) ensures that a flow must enter its destination node completely. Constraint (4.7) ensures that the flow entering a node is equal to the flow that exits that specific node.

4.3 Performance Measures

4.3.1 Financial

The profit of a supplier in any business transaction is defined as the cost to the supplier (to deliver its good/service) subtracted from the revenue (user cost) generated from the sale of the goods/services. This means that when a supplier wishes to increase its profit, it must merely either increase revenue or decrease the cost of supplying the product. All of this must optimally take place while maintaining the same level of quality of the product since a decrease in product quality will most likely lead to a loss of consumer support and loyalty. Thus the financial gain that is used as a performance measure in this dissertation will focus on a decrease in supplier cost, and not growth in generated revenue. If the supplier cost is decreased, even just for certain periods of time, the supplier profit is increased without a change in product quality.

4.3.2 Optimality

To ensure the best use of the available resources optimality is an important consideration in this dissertation. This performance measure is in place to ensure that the principles of optimal resource allocation are followed and implemented by the solution. If the optimality of the resource allocation can be tested through calculation as well as computation, it can be shown to be successful.

4.3.3 Scalability

For the model to be deemed scalable, it must execute and produce a solution when implemented on small topologies or problem sets as well as when implemented on larger problem set-ups. The scalability is tested through the use of the other performance measures. If the solution is considered accurate on a small scale, and still achieves a solution when larger problem data is introduced (without an unreasonable execution time) the approach can be considered scalable in that scenario. An unreasonable execution time would be such that an operator implementing the software would notice the difference, thus to human evaluation, it must remain consistent. Because this dissertation uses an offline approach, with the aim of implementation to users that want a VN for certain intervals of set time (determined by a SP) this is an acceptable measure.

4.4 Chapter Summary

Using the information gathered through the literature review a model is formulated for this dissertation. This specific model is aimed to solve the problem posed for this study. Performance measures are proposed to verify and validate this proposed solution, as well as a strategy to test the validity of this model.

Chapter 5

Computational Results

In this chapter various computations are completed in order to verify and validate the model. This is completed using the methodology discussed in the previous chapter. The experimental set-up and information pertaining to the tests are discussed. Comparisons are made between computations to illustrate results and improvements.

5.1 Experimental Setup

In this section, some terms and structures are discussed to ensure clarity. These are the parameters that are used in this experiment and must be defined in order to classify this approach.

5.1.1 Network Setup

The substrate network and virtual network are identical in order to ensure simplicity of understanding when considering the true goal of this dissertation. This dissertation is not focussed on the optimal conversion of physical resources into a virtual network structure that can be used to embed various virtual networks.

For this reason, the physical infrastructure is not discussed in detail, and it is assumed that the given virtual network structure is the result of the proper conversion of underlying physical resources. In this virtual network structure, two components are significant when considering the resource allocation process. These are nodes and links.

Nodes:
These are the nodes of the network, and when referring to the node capacity, this is the processing power or storage space that this node can provide at its maximum. This is where the software is stored, and the functions are virtualized.

Links:
The links are the connection between two nodes, in order to establish communication. In reality, these links can be physical wires or wireless forms of connection. These links transfer data between connected nodes, and this is what this dissertation refers to as link capacity, which is synonymous with bandwidth.

The commodity is another term that is used in this experiment. A commodity is a set of communicating nodes with links connecting them, which includes data such as the node and link capacity, and can also be used when considering network demands. A commodity for a network demand will contain information such as node and link demands and flow direction.

5.1.2 Requests

A VNR is a request of resources needed to host a VN by a user. VNRs will be perceived by the system in the following manner:

Node demand:
This is the capacity that the user needs from a virtual node, which can be CPU capacity

or processing power among others. This capacity demand is for the use of the available node capacity.

Link demand:
The demand for a link is the bandwidth required by the user in order to ensure effective communications between the virtual nodes. This is needed based on the functions that are needed by the various nodes. If the links between nodes are not correct the processing time needed to retrieve all the necessary functions needed by the user will be drastically increased, which is not the satisfactory desired outcome.

Flow direction:
There are two forms of data transfer in this context: bi-directional and uni-directional data transfer. The flow capacity of this network setup is bi-directional, while the VN request flow is only in one direction.

Monetary value:
Each request has a certain financial gain associated with that specific request; this financial gain is also coupled with the different scenarios for which the request can be placed. From the perspective of the supplier of the virtual network (a ISP for example), the financial gain is calculated by the cost per bandwidth that the user will pay during a user's request minus the cost to offer that service. A certain profit is also linked to accepting a network request, which varies by the urgency of the request and the number of resources needed. For this study, the cost per capacity of the supplier of the infrastructure will be simplified, as well as the cost of the user per demand. This is done in order to easily show the effect of SRA on the financial gain in virtual network embedding. These values are shown in table 5.1.

Table 5.1: Financial Information

Function	**Cost to User (Revenue)**	**Cost to Supplier**
Nodes	30	20
Links	30	20

5.1.3 Scenarios

Scenarios are simply versions of requests where the demand of the VNR differs.

Each user's request has specific scenarios coupled with that specific request to enable SRA. In this experimental set-up, there will be different scenarios for each request in each case study. The real world data for these scenarios will be compiled using historical data of specific requests made by specific users. Each scenario has specific differences considering other scenarios in the experiment. These differences can be in the form of node and link demands. This, in turn, causes the monetary value of that request to shift, per scenario.

In reality, it cannot be assumed that one scenario will be a maximum scenario since another scenario can have specific demands that are higher than the scenario perceived as the maximum scenario, this is the reason why historical data plays an important role when considering real-world applications. This will be compared to an experimental set-up of the same nature, but which has no scenarios in order to compare the financial gain achieved over existing methods that do not use SRA approaches.

If the user still pays for their requested resources, but the supplier can use these scenarios to use fewer resources during certain situations, the overall profit of the supplier can be increased through various methods. For comparative purposes, it is assumed that for each request the different scenarios are used for certain portions of the embedding time and shall be discussed for each case study.

5.1.4 Supplier

The supplier, in this case, is discussed in detail in section 2.6. The supplier provides the actual resources needed to host a VN. The supplier provides a product for consumption to consumers.

5.1.5 User

The user is the client that the VN is created for and can be an individual, business or other entity. The user makes a VNR based on their specific needs and is billed or charged accordingly by the supplier. When referring to user cost or revenue in this dissertation, it is the price that the user pays for the use of the VN that they requested (the product delivered by the supplier).

5.2 Case Study A - Model Verification

5.2.1 Worst-Case

This case study section does not contain stochastic elements and is used for comparative purposes. This is used to test the model formulation purely on the ability to allocate resources optimally when there are several VNRs. In this worst-case version of this case study, the maximum resources that can be demanded by the requests are used as request data, and the maximum load is determined by using the maximum demand over the scenarios that will be available in the stochastic version of this case study.

Network Setup

The virtual network structure that is available for this experiment is shown in figure 5.1. This setup has 10 nodes and 12 links. The simple structure is used to ensure verification by hand is possible and that the solution can easily be checked for optimality.

The node and link capacities are listed in table 5.2 and table 5.3 respectively. These are the available resources that can be used to host a VNR. The financial information about these capacities is also shown.

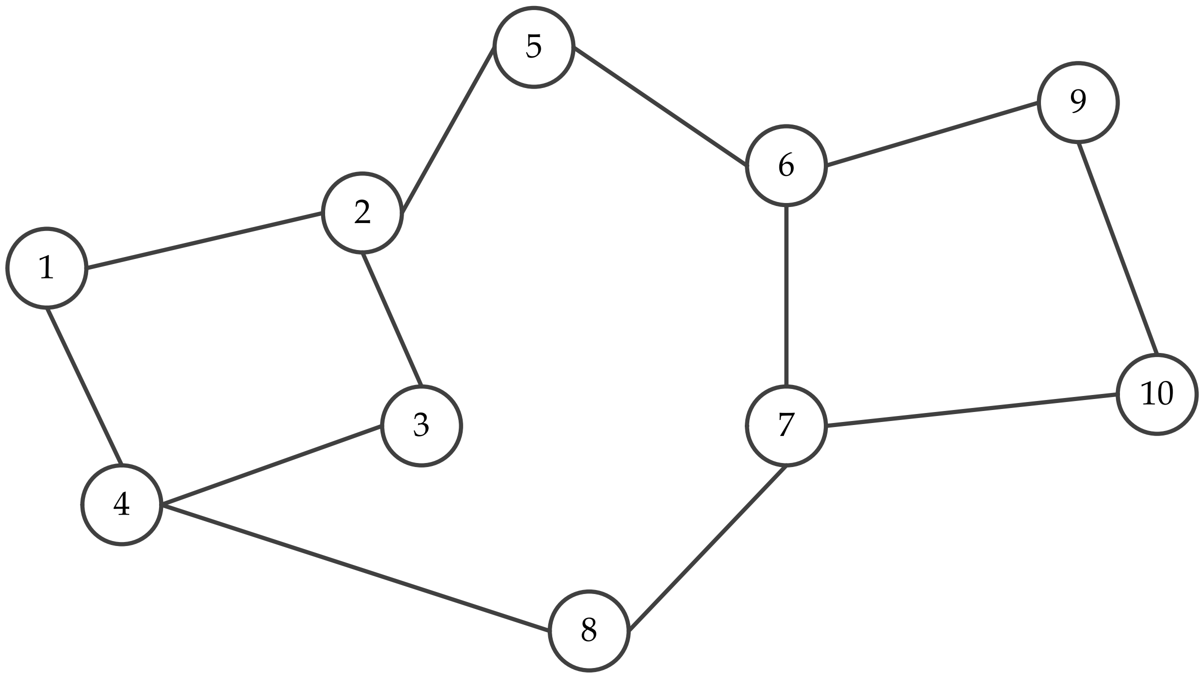

Figure 5.1: Illustration of Case Study A Network Setup

Table 5.2: Node Capacities Case Study A

Node	**Capacity**	**Maximum Cost to Supplier**	**Maximum Cost to User**	**Maximum Profit**
1	5	100	150	50
2	6	120	180	60
3	7	140	210	70
4	8	160	240	80
5	5	100	150	50
6	6	120	180	60
7	7	140	210	70
8	9	180	270	90
9	5	100	150	50
10	6	120	180	60
		Total Profit		640

Table 5.3: Link Capacities Case Study A

Link Pair	Total Capacity	Directional Capacity	Maximum Cost to Supplier	Maximum Cost to User	Maximum Profit
(1,2)	20	10	200	300	100
(1,4)	28	14	280	420	140
(2,3)	24	12	240	360	120
(3,4)	30	15	300	450	150
(2,5)	28	14	280	420	140
(4,8)	24	12	240	360	120
(5,6)	30	15	300	450	150
(8,7)	26	13	260	390	130
(6,7)	20	10	200	300	100
(6,9)	24	12	240	360	120
(7,10)	24	12	240	360	120
(9,10)	26	13	260	390	130
		Total	**Profit**		1520

Requests

There are three separate requests in this case study, labelled as request 1 (R1), request 2 (R2) and request 3 (R3). These requests are shown in figure 5.2. The node and link demands for these requests are given in table 5.4 and 5.5 respectively with the maximum possible profit that can be achieved from these requests also shown for these request demands. The maximum profit is easily calculated since only the maximum load/demand is used for the whole time the request is embedded. The calculation is then just the revenue minus the cost to supply that service.

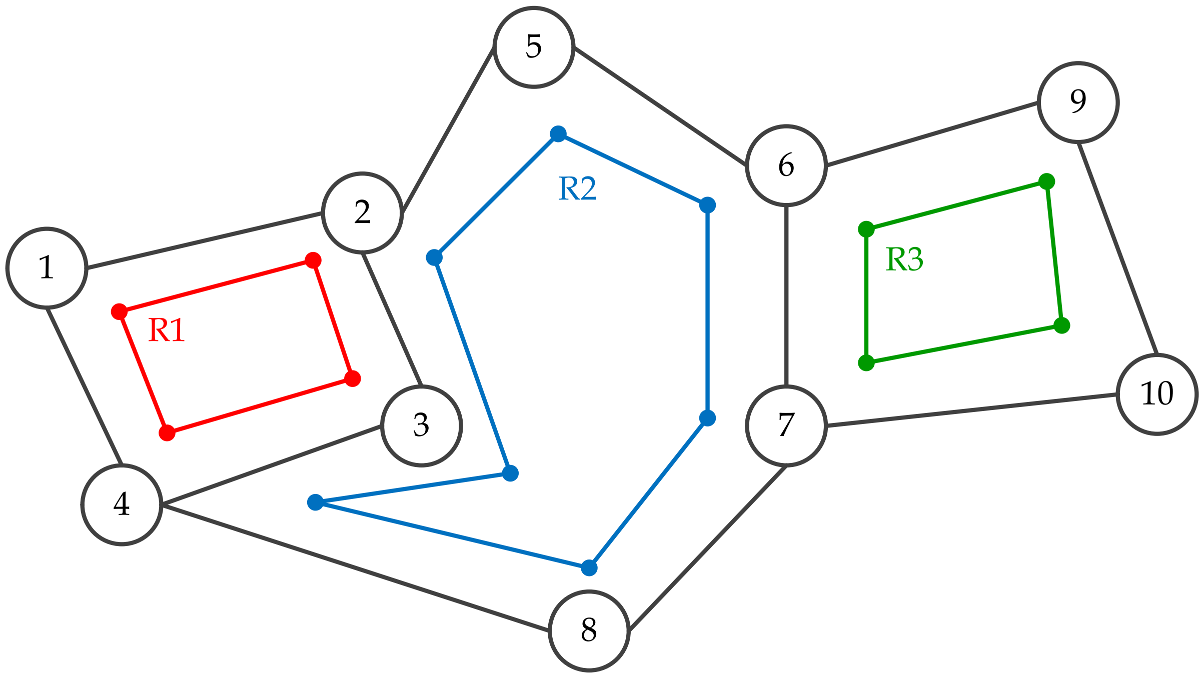

Figure 5.2: Illustration of Requests for Case Study A

Table 5.4: Node Request Demands Case Study A: Worst Case

Request (R)	**Node**	**Node Demand**	**Maximum Possible Profit**
1	1	3	30
	2	5	50
	3	4	40
	4	5	50
2	2	4	40
	3	5	50
	4	7	70
	5	4	40
	6	5	50
	7	5	50
	8	6	60
3	6	4	40
	7	6	60
	9	4	40
	10	5	50

Table 5.5: Link Request Demands Case Study A: Worst Case

Request (R)	Link Direction	Link Demand	Cost to Supplier	Cost to User	Possible Profit
1	(1,2)	9	180	270	90
	(2,3)	10	200	300	100
	(3,4)	8	160	240	80
	(4,1)	11	220	330	110
2	(2,5)	10	200	300	100
	(5,6)	12	240	360	120
	(6,7)	8	160	240	80
	(7,8)	11	220	330	110
	(8,4)	10	200	300	100
	(4,3)	12	240	360	120
	(3,2)	10	200	300	100
3	(6,9)	10	200	300	100
	(9,10)	12	240	360	120
	(10,7)	9	180	270	90
	(7,6)	8	160	240	80

Scenarios

There are not any scenarios in this case study, the maximum node and link demand for each request are used. These maximum demands will be the maximum demand over the possible scenarios in the stochastic version of the case study.

Result

The data is used as input, and the algorithm is executed on this experimental setup. The solution achieved from the embedding algorithm is given in the form of which requests are placed using a binary value, as discussed in 4.2. The results for this setup is shown in table 5.6.

Table 5.6: Case Study A: Worst Case Embedding Results

Request Number	**Placement Result 1=Accepted, Otherwise 0**
1	1
2	0
3	1

Accepting request 1 and request 3 is provided as part of the optimal solution. Request 2 is effectively rejected since the substrate infrastructure cannot host all the requests at the same time.

The financial information for this embedding solution is summarised in table 5.7. Firstly the total user cost (revenue) for each request is shown. Secondly, the total supplier cost is shown for each request. Lastly, the profit for each of these requests is shown, which is achieved by subtracting the supplier cost from the generated revenue. In this simple form, it can be seen that the highest profit from a single request is from request 2. However, upon further inspection, it is seen that the combined profit of request 1 and request 3 is higher than that of request 2 on its own. This shows the optimality of the solution and that the highest possible profit is achieved by the algorithm.

Table 5.7: Case Study A Financial Information Worst Case

Total User Cost		
Request 1	**Request 2**	**Request 3**
1650	3270	1740
Total Supplier Cost		
Request 1	**Request 2**	**Request 3**
1100	2180	1160
Available Profit		
Request 1	**Request 2**	**Request 3**
550	1090	580

5.2.2 Stochastic

This is the implementation of the stochastic approach to the above case study. Scenarios are included in this set-up. This experiment is essentially similar to the worst-case version as discussed in section 5.2.1 but is used to show the improvement that can be gained from SRA techniques.

Network Setup

The virtual network structure that is available for this experiment is shown in figure 5.1. Note the structure is not changed from the worst case version of the case study. The node and link capacities are listed in table 5.2 and table 5.3 and also remain the same as in section 5.2.1. The node and link capacities do not change when considering the worst-case and stochastic versions of this case study. The capacity of the VN remains constant, and the addition of scenarios considering demand is implemented. This is done to achieve a comparable result.

Requests

There are three separate requests in this set-up, labelled as request 1 (R1), request 2 (R2) and request 3 (R3). These requests are shown in figure 5.2. The request setups remain constant as well.

Scenarios

There are two scenarios for each separate request in this set-up. The node and link demands for these requests per scenario are given in table 5.8 and table 5.9 respectively with the maximum possible profit that can be achieved from these requests also shown for these request demands. Note that this maximum profit changes from scenario to scenario.

Table 5.8: Node Request Demands Case Study A: Stochastic

Request (R)	**Scenario (S)**	**Node**	**Node Demand**	**Possible Profit**
1	1	1	3	30
		2	5	50
		3	2	20
		4	4	40
1	2	1	3	30
		2	3	30
		3	4	40
		4	5	50
2	1	2	4	40
		3	5	50
		4	7	70
		5	4	40
		6	3	30
		7	3	30
		8	2	20
2	2	2	4	40
		3	3	30
		4	2	20
		5	4	40
		6	5	50
		7	5	50
		8	6	60
3	1	6	4	40
		7	6	60
		9	2	20
		10	3	30
3	2	6	2	20
		7	4	40
		9	4	40
		10	5	50

Table 5.9: Link Request Demands Case Study A: Stochastic

Request (R)	Scenario (S)	Link Direction	Link Demand	Cost to Supplier	Cost to User	Possible Profit
1	1	(1,2)	9	180	270	90
		(2,3)	10	200	300	100
		(3,4)	4	80	120	40
		(4,1)	5	100	150	50
1	2	(1,2)	3	60	90	30
		(2,3)	3	60	90	30
		(3,4)	8	160	240	80
		(4,1)	11	220	330	110
2	1	(2,5)	6	120	180	60
		(5,6)	12	240	360	120
		(6,7)	8	160	240	80
		(7,8)	2	40	60	20
		(8,4)	10	200	300	100
		(4,3)	12	240	360	120
		(3,2)	10	200	300	100
2	2	(2,5)	10	200	300	100
		(5,6)	6	120	180	60
		(6,7)	5	100	150	50
		(7,8)	11	220	330	110
		(8,4)	2	40	60	20
		(4,3)	7	140	210	70
		(3,2)	4	80	120	40
3	1	(6,9)	10	200	300	100
		(9,10)	12	240	360	120
		(10,7)	9	180	270	90
		(7,6)	5	100	150	50
3	2	(6,9)	7	140	210	70
		(9,10)	2	40	60	20
		(10,7)	3	60	90	30
		(7,6)	8	160	240	80

Result

The data is used as input, and the algorithm is executed on this experimental setup. The solution achieved from the embedding algorithm is given in the form of which requests are placed using a binary value, as discussed in section 4.2. The results for this setup is shown in table 5.10.

Table 5.10: Case Study A: Stochastic Embedding Results

Request Number	**Placement Result 1=Accepted, Otherwise 0**
1	1
2	0
3	1

As is seen request 1 and request 3 is embedded in this solution. Request 2 is effectively rejected since the substrate infrastructure cannot host all the requests at the same time.

For comparative purposes, it is assumed that for each request the first scenario (scenario 1) is actively used 60% of the time that the VN is embedded and the second scenario (scenario 2) is used for the remaining 40% of the VN embedding time. This is referred to as the 60/40 division for case study A and is used later in this chapter for comparative purposes.

The financial information for this embedding solution is summarised in table 5.11. Firstly the total user cost (revenue) for each request is shown. Secondly, the total supplier cost is shown for each request.

Table 5.11: Case Study A Total User Cost and Total Supplier Cost Stochastic

	Total User Cost		
	Request 1	**Request 2**	**Request 3**
Scenario 1	1260	2640	1530
Scenario 2	1200	2220	1050
	Total Supplier Cost		
	Request 1	**Request 2**	**Request 3**
Scenario 1	840	1760	1020
Scenario 2	800	1480	700

5.3 Case Study A Comparison- Model Verification Worst-Case vs Stochastic

This comparison is aimed at case study A and will compare the stochastic and worst case version of this case study, as completed earlier in this chapter. In table 5.12 the VN acceptance is compared. In this case study, the same requests are accepted by the algorithm in the stochastic and worst case version of the problem.

Table 5.12: Case Study A: Embedding Results Comparison

Request Number	Worst Case Acceptance	Stochastic Acceptance
1	1	1
2	0	0
3	1	1

The financial information for the embedding is given in table 5.13 and table 5.14 and shows the difference in revenue and supplier cost when considering scenarios for each request, as well as the worst case maximum load over each request.

Table 5.13: Case Study A Total User Cost Comparison

	Total User Cost		
	Request 1	**Request 2**	**Request 3**
Scenario 1	1260	2640	1530
Scenario 2	1200	2220	1050
Maximum Scenario	1650	3270	1740

Table 5.14: Case Study A Total Supplier Cost Comparison

	Total Supplier Cost		
	Request 1	**Request 2**	**Request 3**
Scenario 1	840	1760	1020
Scenario 2	800	1480	700
Maximum Scenario	1100	2180	1160

In table 5.15 the financial information of case study A is given with a specific comparative focus on the stochastic improvement that is offered by the algorithm. In the first part of the table, the profit is discussed for the worst case scenario in which no SRA techniques are used. In the second part of the table, the profit achieved through the use of SRA is shown using the 60/40 division discussed in section 5.2.2. The last column of the table illustrated the improvement in profit for the supplier when using the SRA techniques.

Table 5.15: Case Study A Profit Summary

	Non-Stochastic Profit			**Stochastic Profit**				**Comparison**
Req	**User Cost**	**Supplier Cost**	**Profit Normal**	**Scen**	**Usage %**	**Supplier Cost**	**Profit**	**Difference Stochastic**
1	-	-	-	1	60	504	-	-
	-	-	-	2	40	320	-	-
	1650	1100	550	*Total*	-	824	826	33.41%
2	-	-	-	1	60	1056	-	-
	-	-	-	2	40	592	-	-
	3270	2180	1090	*Total*	-	1648	1622	32.8%
3	-	-	-	1	60	612	-	-
	-	-	-	2	40	280	-	-
	1740	1160	580	*Total*	-	892	848	31.6%

From this table, the improvements per request can be seen when using SRA techniques, although only the requests that are accepted by the system will experience these financial improvements since the other requests are effectively rejected, and thus no revenue can be gained from these sources. In table A.1 the overall average financial improvement of SRA in case study A is shown. From this information, it is seen that an overall stochastic improvement of 32.5% is achieved through the use of SRA in this case study. Even though the same requests are accepted in each version of this case study, the financial gain can be achieved by the supplier by using these techniques.

This comparison illustrates the improvement that can be achieved by merely using SRA elements in the VNE approach. This is the improvement for the accepted requests and does not consider the rejected requests. The financial comparison of case study A is shown in figure 5.3. Each request is shown with the user cost (the income that is gained from the client), the supplier cost (the cost to host that VNR for the supplier) and the profit that the supplier can make. The income from the user does not change, but the profit and supplier cost is shown for non-stochastic allocation as well as stochastic allocation, which shows the increase in profit for the supplier possible from the use of SRA.

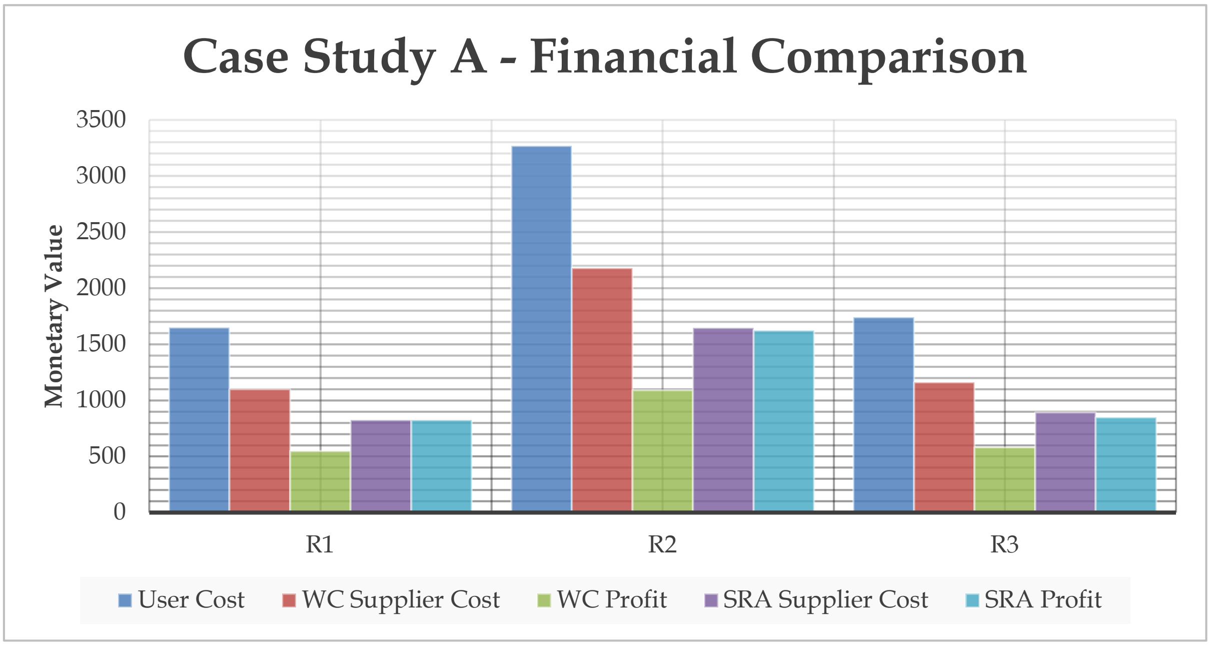

Figure 5.3: Financial Comparison of Case Study A

In this figure, it is seen that if the supplier cost can be decreased, even just for certain periods of time, the profit margin can be increased.

5.4 Case Study B - Model Validation

5.4.1 Worst-Case

This version of the case study is used in order to compare the proposed model to current works, which are similar to the worst case version of this case study. This section does not use stochastic elements. This is used to test the model formulation purely on the ability to optimally allocate resources when there are several VNRs. In this worst-case version of this case study, the maximum resources that can be demanded by the requests are used as request data, and this maximum load is determined by using the maximum demand over the scenarios that will be available in the stochastic version of this specific case study.

Network Setup

The virtual network structure that is available for this experiment is shown in figure 5.4. This setup has 20 nodes and 30 links.

The node and link capacities are listed in table 5.16 and table 5.17 respectively. These are the available resources that can be used to host a VNR. The financial information about these capacities is also shown.

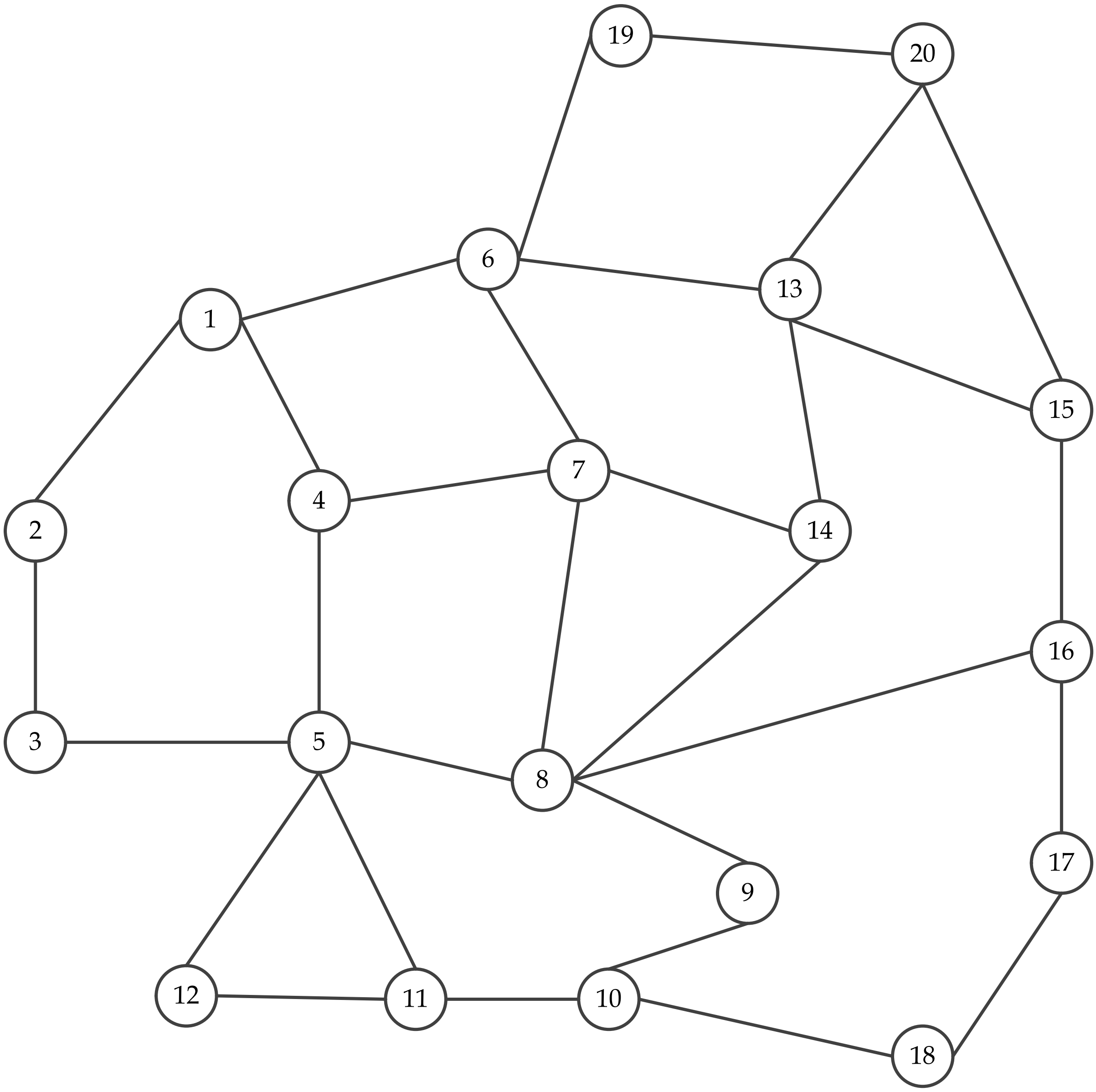

Figure 5.4: Illustration of Case Study B Network Setup

Table 5.16: Node Capacities Case Study B: Worst Case

Node	**Capacity**	**Maximum Cost to Supplier**	**Maximum Cost to User**	**Maximum Profit**
1	15	300	450	150
2	12	240	360	120
3	13	260	390	130
4	20	400	600	200
5	19	380	570	190
6	16	320	480	160
7	20	400	600	200
8	18	360	540	180
9	14	280	420	140
10	12	240	360	120
11	11	220	330	110
12	10	200	300	100
13	17	340	510	170
14	18	360	540	180
15	15	300	450	150
16	14	280	420	140
17	12	240	360	120
18	10	200	300	100
19	13	260	390	130
20	16	320	480	160
		Total Profit		2950

Table 5.17: Link Capacities Case Study B

Link Pair	**Total Capacity**	**Directional Capacity**	**Maximum Cost to Supplier**	**Maximum Cost to User**	**Maximum Profit**
(1,2)	34	17	340	510	170
(1,4)	48	24	480	720	240
(1,6)	46	23	460	690	230
(2,3)	30	15	300	450	150
(3,5)	36	18	360	540	180
(4,5)	52	26	520	780	260
(4,7)	54	27	540	810	270
(5,8)	40	20	400	600	200
(5,11)	38	19	380	570	190
(5,12)	40	20	400	600	200
(6,7)	36	18	360	540	180
(6,13)	40	20	400	600	200
(6,19)	30	15	300	450	150
(7,8)	30	15	300	450	150
(7,14)	24	12	240	360	120
(8,9)	36	18	360	540	180
(8,14)	60	30	600	900	300
(8,16)	50	25	500	750	250
(9,10)	48	24	480	720	240
(11,10)	32	16	320	480	160
(12,11)	52	26	520	780	260
(13,14)	56	28	560	840	280
(13,15)	34	17	340	510	170
(13,20)	52	26	520	780	260
(15,16)	44	22	440	660	220
(16,17)	42	21	420	630	210
(17,18)	56	28	560	840	280
(18,10)	36	18	360	540	180
(19,20)	42	21	420	630	210
(20,15)	32	16	320	480	160
			Total	**Profit**	6250

Requests

There are 14 separate requests in this case study. These requests are shown in figure 5.5. The node and link demands for these requests are given in table A.3 and table A.4 respectively with the maximum possible profit that can be achieved from these requests.

The maximum profit is easily calculated since only the maximum load/demand is used for the whole time the request is embedded. The calculation is then the cost to supply the service subtracted from the generated revenue.

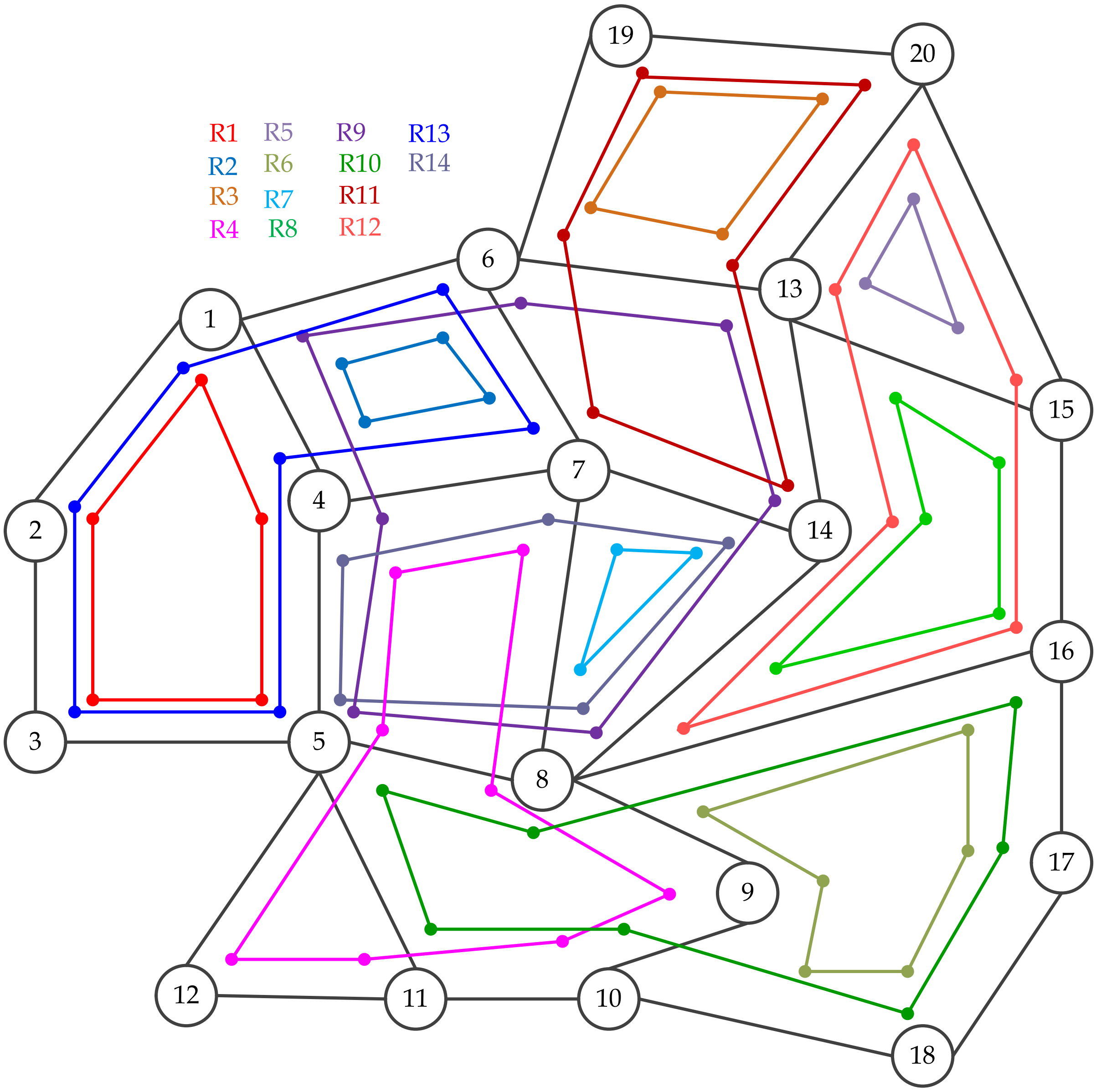

Figure 5.5: Illustration of Requests for Case Study B

Scenarios

There are not any scenarios in this case study section, the maximum node and link demand for each request are used. These maximum demands will be the maximum demand over the possible scenarios in the stochastic version of the case study. This is purely for comparative purposes.

Result

The data is used as input, and the algorithm is executed on this experimental setup. The solution achieved from the embedding algorithm is given in the form of which requests are placed using a binary value, as discussed in section 4.2. The results for this setup is shown in table 5.18.

Table 5.18: Case Study B: Worst Case Embedding Results

Request Number	**Placement Result** **1=Accepted, Otherwise 0**
1	1
2	1
3	0
4	1
5	0
6	0
7	0
8	1
9	0
10	0
11	1
12	1
13	0
14	0

Six requests are accepted, and the rest are effectively rejected since the substrate infrastructure cannot host all the requests at the same time.

The financial information for this embedding solution is summarised in table A.5.

Firstly the total user cost (revenue) for each request is shown. Secondly, the total supplier cost is shown for each request. Lastly, the profit for each of these requests is shown, which is achieved by subtracting the supplier cost from the generated revenue.

5.4.2 Stochastic

This is the implementation of the stochastic approach to this case study. Scenarios are included in this set-up. This experiment is essentially similar to the worst-case version as discussed in section 5.4.1 but is used to show the improvement that can be gained from SRA techniques.

Network Setup

The virtual network structure that is available for this experiment is shown in figure 5.4. Note the structure is not changed from the worst case version of the case study. The node and link capacities are listed in table 5.16 and table 5.17 and also remain the same as in section 5.4.1. The node and link capacities do not change when considering the worst-case and stochastic versions of this case study. The capacity of the VN remains constant, and the addition of scenarios considering demand is implemented.

Requests

There are 14 separate requests in this case study. These requests are shown in figure 5.5 and are the same as used in section 5.4.1.

Scenarios

There are 3 scenarios for each separate request in this case study. The node and link demands for these requests per scenario are given in table A.7 and table A.8 respectively with the maximum possible profit that can be achieved from these requests also shown for these request demands. Note that this maximum profit changes from scenario to scenario.

Result

The data is used as input, and the algorithm is executed on this experimental setup. The solution achieved from the embedding algorithm is given in the form of which requests are placed using a binary value, as discussed in section 4.2. The results for this setup is shown in table 5.19.

Table 5.19: Case Study B: Stochastic Embedding Results

Request Number	Placement Result 1=Accepted, Otherwise 0
1	1
2	1
3	1
4	1
5	0
6	0
7	1
8	1
9	0
10	0
11	0
12	1
13	0
14	0

Seven requests are accepted, and the rest are effectively rejected since the substrate infrastructure cannot host all the requests at the same time.

For comparative purposes, it is assumed that for each request the first scenario (scenario 1) is actively used 30% of the time that the VN is embedded, the second scenario (scenario 2) is used for 40% of the VN embedding time and the third scenario (scenario 3) is used for the remaining 30% of the VN embedding time.

This is referred to as the 30/40/30 division for case study B.

The financial information for this embedding solution is summarised in table A.6. Firstly the total user cost (revenue) for each request is shown. Secondly, the total supplier cost is shown for each request.

5.5 Case Study B Comparison- Worst-Case vs Stochastic

This comparison is aimed as case study B and will compare the stochastic and worst case version of this case study, as completed earlier in this chapter. In table 5.20 the VN acceptance is compared. In this case study, the same requests are not accepted by the algorithm in the stochastic and worst case version of the problem. As can be seen from the table, the stochastic version accepts one more request than the worst case version. Furthermore, other requests are accepted of rejected in each version of this case study.

Table 5.20: Case Study B: Embedding Results Comparison

Request Number	**Worst Case Accepted**	**Stochastic Accepted**
1	1	1
2	1	1
3	0	1
4	1	1
5	0	0
6	0	0
7	0	1
8	1	1
9	0	0
10	0	0
11	1	0
12	1	1
13	0	0
14	0	0

The financial information for the embedding is given in table A.9 and table A.10 and shows the difference in revenue and supplier cost when considering scenarios for each request, as well as the worst case maximum load over each request.

In table 5.21, the financial information of case study B, is given, with a specific comparative focus on the stochastic improvement that is offered by the algorithm. In the first part of the table, the profit is discussed for the worst case scenario in which no SRA techniques are used. In the second part of the table, the profit achieved through the use of SRA is shown using the 30/40/30 division discussed in section 5.4.2. The last column of the table illustrated the improvement in profit for the supplier when using the SRA techniques.

	Non-Stochastic Profit			**Stochastic Profit**				**Comparison**
Req	**User Cost**	**Supplier Cost**	**Profit Normal**	**Scen**	**Usage %**	**Supplier Cost**	**Profit**	**Difference Stochastic**
1	-	-	-	1	30	372	-	-
	-	-	-	2	40	480	-	-
	-	-	-	3	30	336	-	-
	2070	1380	690	*Total*	-	1188	882	21.769%
2	-	-	-	1	30	456	-	-
	-	-	-	2	40	504	-	-
	-	-	-	3	30	414	-	-
	2430	1620	810	*Total*	-	1374	1056	23.295%
3	-	-	-	1	30	372	-	-
	-	-	-	2	40	488	-	-
	-	-	-	3	30	306	-	-
	1800	1200	600	*Total*	-	1166	634	5.363%
4	-	-	-	1	30	714	-	-
	-	-	-	2	40	912	-	-
	-	-	-	3	30	678	-	-
	3960	2640	1320	*Total*	-	2304	1656	20.29%
5	-	-	-	1	30	288	-	-
	-	-	-	2	40	376	-	-
	-	-	-	3	30	264	-	-
	1560	1040	520	*Total*	-	928	632	17.722%
6	-	-	-	1	30	600	-	-
	-	-	-	2	40	792	-	-
	-	-	-	3	30	546	-	-
	3450	2300	1150	*Total*	-	1938	1512	23.942 %
7	-	-	-	1	30	168	-	-
	-	-	-	2	40	216	-	-
	-	-	-	3	30	204	-	-
	1080	720	360	*Total*	-	588	492	26.829 %
8	-	-	-	1	30	390	-	-
	-	-	-	2	40	528	-	-

	-	-	-	3	30	360	-	-
	2220	1480	740	*Total*	-	1278	942	21.444 %
	-	-	-	1	30	408	-	-
9	-	-	-	2	40	544	-	-
	-	-	-	3	30	378	-	-
	2370	1580	790	*Total*	-	1330	1040	24.038 %
	-	-	-	1	30	546	-	-
10	-	-	-	2	40	688	-	-
	-	-	-	3	30	480	-	-
	3300	2200	1100	*Total*	-	1714	1586	30.643%
	-	-	-	1	30	396	-	-
11	-	-	-	2	40	520	-	-
	-	-	-	3	30	384	-	-
	2340	1560	780	*Total*	-	1300	1040	25%
	-	-	-	1	30	450	-	-
12	-	-	-	2	40	464	-	-
	-	-	-	3	30	378	-	-
	2490	1660	830	*Total*	-	1292	1198	30.718 %
	-	-	-	1	30	492	-	-
13	-	-	-	2	40	664	-	-
	-	-	-	3	30	540	-	-
	3270	2180	1090	*Total*	-	1696	1574	30.75%
	-	-	-	1	30	282	-	-
14	-	-	-	2	40	408	-	-
	-	-	-	3	30	342	-	-
	1770	1180	590	*Total*	-	1032	738	20.054 %

Table 5.21: Case Study B Profit Summary

From this table, the improvements per request can be seen when using SRA techniques, although only the requests that are accepted by the system will experience these financial improvements since the other requests are effectively rejected, and thus no revenue can be gained from these sources.

This improvement in profit is illustrated in figure 5.6. This is the improvement in profit that the supplier can gain from the SRA approach per request.

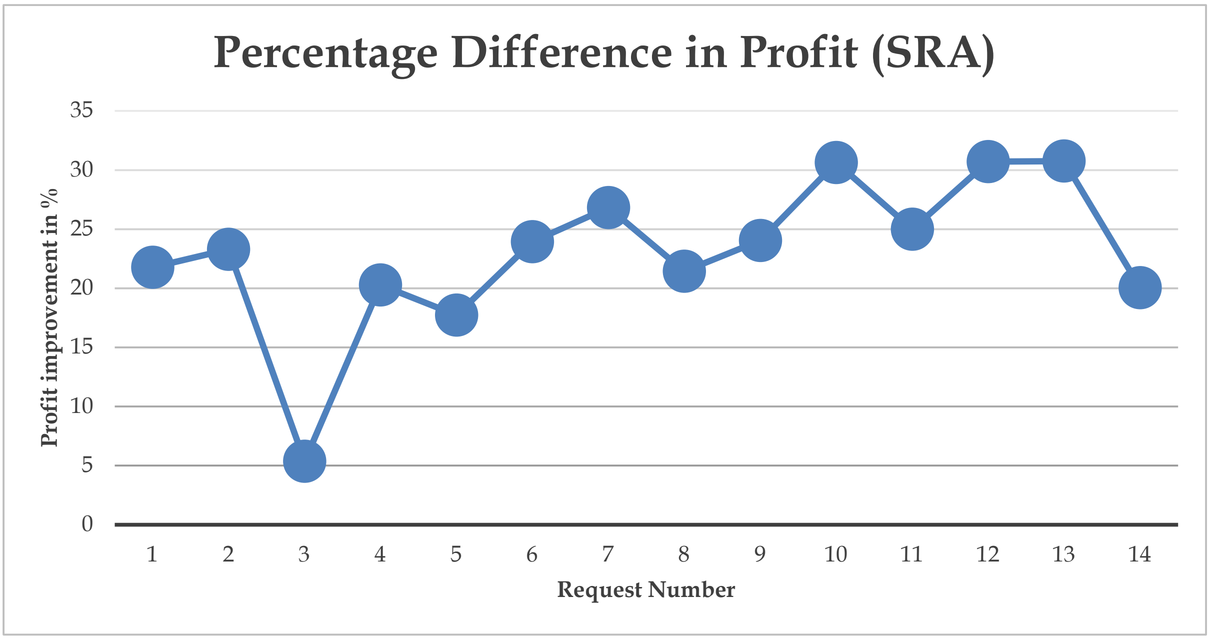

Figure 5.6: Stochastic Improvement of Profit of Case Study B

From this figure, it is seen that the improvement can fluctuate with the different requests. The underlying structure, request setup and request demands can all play a role in the improvement that stochastic allocation can offer. Nevertheless, the average improvement is seen as more than 5% at a minimum, which is favourable in large-scale environments. In table A.2, the overall average financial improvement of SRA in case study B, is shown and calculated as 24.64%. This is the improvement for the accepted requests and does not consider the rejected requests.

This comparison illustrates the improvement that can be achieved by merely using SRA elements in the VNE approach. The financial comparison of case study B is shown in figure 5.7. Each request is shown with the user cost (the income that is gained from the client), the supplier cost (the cost to host that VNR for the supplier) and the profit that the supplier can make. The income from the user does not change, but the profit and supplier cost is shown for non-stochastic allocation as well as stochastic allocation, which shows the increase in profit for the supplier possible from the use of SRA.

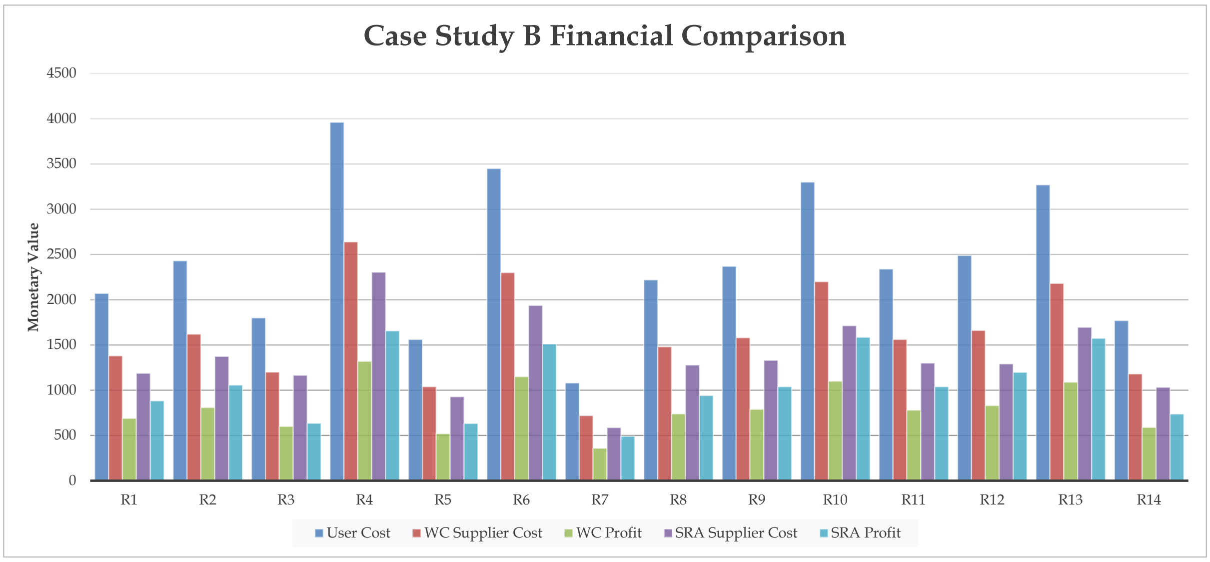

Figure 5.7: Financial Comparison of Case Study B

5.6 Scalability

The comparison of these case studies experimental set-ups is what can be used to ensure verification and validation of the model created for this dissertation.

Case Study A Worst-Case vs Case Study B Worst-Case

The worst case versions of each case study only have a maximum load demand scenario, which means there is only one computation time to evaluate. Thus the results for this is incorporated into the next section. This is done to illustrate these results compared to situations with one or more scenarios.

Case Study A Stochastic vs Case Study B Stochastic

In this comparison, the scalability of the solution must be considered. In figure 5.8 the computation time of case study A and case study B is shown. The maximum worst case versions of the case studies are shown as a data point as one scenario (Max scenario). The other data points are for the stochastic versions of the case studies with one, two or more scenarios. The computation time considering the number of scenarios shows what can be expected as the number of scenarios increase. For case study A the forecasted computation time per number of scenarios appears to be linear. For case study B the forecast increases at a higher rate than case study A. The computation time for the maximum load is also higher than when only one scenario is considered. This can indicate that the substrate infrastructure may also influence computation times.

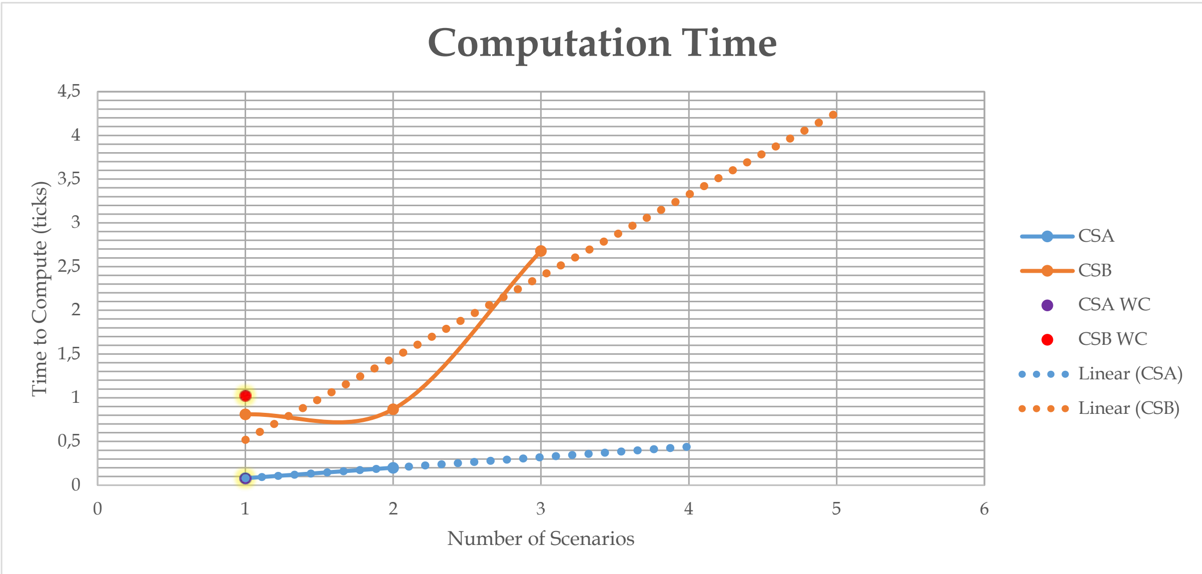

Figure 5.8: Comparison of Computation Time

5.7 Chapter Summary

The model formulation as given in section 4.2 is implemented on two case studies in this chapter. The results are compared to evaluate the stochastic improvement as well as scalability. The model is shown to achieve optimal embedding with improved financial gains.

Chapter 6

Conclusion and Recommendation

This chapter is the final chapter of this dissertation. In this chapter the dissertation conclusion is given, as well as summations of the completed works. Recommendations are made - based on the results of the study- for future work.

The VNE problem is a crucial component in the future architecture of the Internet and virtual environments. An embedding algorithm will receive certain VNRs from users, requesting the use of a certain amount of resources. This demand is evaluated, along with all the other demands from other received VNRs, and assessed. Dedicated algorithms then allocate resources to the chosen VNRs that have been accepted to be hosted. To optimally allocate resources in this environment is a numerical challenge and has been researched in various papers.

A multi-commodity flow approach was used to model the flow in a virtualized network structure. A model was created with the objective of maximising the profit that a supplier can generate, subject to certain constraints. This algorithm is implemented through the use of the CPLEX programming environment.

Two case studies are presented in this dissertation. Case study A is simplistic and is used to ensure calculation by hand is possible. This is for the purpose to verify the model and ensure optimal resource allocation. Case study B is more complicated and is used to illustrate the scalability of the model and validate the results.

This dissertation uses varying scenarios for the case studies. In the practical environment some instances considering scenario based allocation, only have a high demand and low demand scenario. In reality, different scenarios will not be classified into these two specific categories, and this is the reason this dissertation used scenarios which could not be classified as one being higher or more demanding than the other and scenario-based demand varied. Only the maximum load is considered more demanding overall than certain scenarios, and this is used in the worst case version of the embedding since that approach does not use scenarios and is therefore equivalent to previous works.

6.1 Overall Conclusion

Is the proposed model feasible?
The model is proven feasible through the use of the first case study (A). It is shown that the solution obtained is optimal, even when verified by hand calculations. The embedding choices are aimed at achieving an objective, which is accomplished. The objective was to maximise the profit of the supplier.

Does the model scale?
When considering the scalability of this model, there are different factors which must be discussed. Firstly, it is seen that the size of the SN can be an influencing factor and secondly, the number of scenarios does not provide a genuinely linear proportionality in all cases. From this information, it can be concluded that experiments on bigger data sets are required to determine the scalability of the proposed model.

Does the model provide financial improvement?
When considering the effect that SRA can have over a deterministic modelling ap-

proach the conclusion can be made that the use of stochastic optimisation techniques can increase the profits of a ISP as well as the experience of the end users. It is shown that an average of 32.5% savings can be obtained for a case study A and an average of 24.64% savings can be obtained for case study B solely through the use of SRA techniques, which is not used in most existing works.

Comparison to existing works

The comparison of this model to existing works is shown through the use of SRA techniques, which is not commonly found in existing VNE approaches. This means that whenever SRA techniques are applied and improvement is made, that improvement is a betterment on most VNE approaches since those approaches mostly do not use SRA optimisation techniques. Financial improvements, as well as improved resource utilisation found in this study, are thus improvements made upon most commonly used VNE solutions.

Other

Many stochastic scenarios may exist for a single request with node and link demands that differ in range. The advantage that this scenario-based approach has is that during the times when the scenario used by a certain request allows it, another request can be placed on the same infrastructure which would not have been possible if the maximum resources were allocated to the original request. Over-provisioning can also be reduced through the use of optimisation techniques which in turn reduces the physical infrastructure needed to host the same number of virtual networks. Because physical space is a finite resource in our ever-expanding world, this is an advantage that can positively influence this situation and create less waste in the physical environment.

6.2 Recommendation and Future Work

Certain performance measures were not explored in this dissertation approach, such as network security or energy efficiency of the embedding. If the scale of the experiment is widened, these factors can be considered in order to determine the influence this research can have in the field. It is recommended to use real-world data, if it can be obtained, to ensure the physical implementation of this solution.

6.3 Final Thoughts and Future Work

The approach followed in this study had the following situation in mind: a supplier that receives many VNRs and then - through the use of an embedding algorithm - decides to accept or reject these requests, based on predetermined factors. The accepted requests are then hosted on the supplier's infrastructure for revenue, for a period. In this off-line version of an embedding problem, the supplier has more control, and the complexity of the system is reduced. This work can be extended through the creation of smart embedding systems, which would physically implement these embedding choices autonomously at specified time intervals.

Bibliography

[1] A. Fischer, J. F. Botero, M. T. Beck, H. de Meer, and X. Hesselbach, “Virtual network embedding: A survey,” *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 1888–1906, 2013.

[2] C. Besiktas, D. Gozupek, A. Ulas, E. Lokman, O. Ozyurt, and K. Ulusoy, “Flexible bandwidth-based virtual network embedding,” pp. 612–617, 2016.

[3] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee, “Network function virtualization: Challenges and opportunities for innovations: Network and service virtualization,” *IEEE Communications Magazine*, pp. 90–97, 2015.

[4] H. T. Nguyen, A. V. Vu, D. L. Nguyen, H. van Nguyen, M. N. Tran, Q. T. Ngo, T.-H. Truong, T. H. Nguyen, and T. Magedanz, “A generalized resource allocation framework in support of multi-layer virtual network embedding based on sdn,” *Computer Networks*, vol. 92, pp. 251–269, 2015.

[5] E. Amaldi, S. Coniglio, A. M. Koster, and M. Tieves, “On the computational complexity of the virtual network embedding problem,” *Electronic Notes in Discrete Mathematics*, vol. 52, pp. 213–220, 2016.

[6] G. Fan and H. Huang, “Scenario-based stochastic resource allocation with uncertain probability parameters,” *Journal of Systems Science and Complexity*, vol. 30, no. 2, pp. 357–377, 2017.

[7] M. Chowdhury, M. R. Rahman, and R. Boutaba, “Vineyard: Virtual network em-

bedding algorithms with coordinated node and link mapping," *IEEE/ACM Transactions on Networking*, vol. 20, no. 1, pp. 206–219, 2012.

[8] G. Sun, H. Yu, L. Li, V. Anand, Y. Cai, and H. Di, "Exploring online virtual networks mapping with stochastic bandwidth demand in multi-datacenter," *Photonic Network Communications*, vol. 23, no. 2, pp. 109–122, 2012.

[9] M. Melo, S. Sargento, U. Killat, A. Timm-Giel, and J. Carapinha, "Optimal virtual network embedding: Energy aware formulation," *Computer Networks*, vol. 91, pp. 184–195, 2015.

[10] G. Tychogiorgos and K. K. Leung, "Optimization-based resource allocation in communication networks," *Computer Networks*, vol. 66, pp. 32–45, 2014.

[11] Y. Li and M. Chen, "Software-defined network function virtualization: A survey," *IEEE Access*, vol. 3, pp. 2542–2553, 2015.

[12] S. D'Oro, L. Galluccio, P. Mertikopoulos, G. Morabito, and S. Palazzo, "Auction-based resource allocation in openflow multi-tenant networks," *Computer Networks*, vol. 115, pp. 29–41, 2017.

[13] T. Wang and M. Hamdi, "Presto: Towards efficient online virtual network embedding in virtualized cloud data centers," *Computer Networks*, vol. 106, pp. 196–208, 2016.

[14] S. U. Sharma and Y. B. Gandole, Eds., *Virtualization Approach to Reduce Network Latency for Thin Client Performance Optimization in Cloud Computing Environment*. Piscataway, NJ and Piscataway, NJ: IEEE, 2014.

[15] G. Chochlidakis and V. Friderikos, Eds., *Robust Virtual Network Embedding for Mobile Networks*. Piscataway, NJ: IEEE, 2015.

[16] O. Kosak, G. Anders, F. Siefert, and W. Rief, Eds., *An Approach to Robust Resource Allocation in Large-Scale Systems of Systems*. IEEE, 2015.

[17] S. Coniglio, A. Koster, and M. Tieves, "Data uncertainty in virtual network embedding: Robust optimization and protection levels," *Journal of Network and Systems Management*, vol. 24, no. 3, pp. 681–710, 2016.

[18] Y. Liang and S. Zhang, "Embedding parallelizable virtual networks," *Computer Communications*, vol. 102, pp. 47–57, 2017.

[19] C. Beşiktaş, D. Gözüpek, A. Ulaş, and E. Lokman, "Secure virtual network embedding with flexible bandwidth-based revenue maximization," *Computer Networks*, vol. 121, pp. 89–99, 2017.

[20] S. Gong, J. Chen, C. Huang, Q. Zhu, and S. Zhao, "Virtual network embedding through security risk awareness and optimization," *KSII Transactions on Internet and Information Systems*, vol. 10, no. 8, 2016.

[21] X. Hesselbach, J. R. Amazonas, S. Villanueva, and J. F. Botero, "Coordinated node and link mapping vne using a new paths algebra strategy," *Journal of Network and Computer Applications*, vol. 69, pp. 14–26, 2016.

[22] M. Feng, J. Liao, S. Qing, T. Li, and J. Wang, "Cove: Co-operative virtual network embedding for network virtualization," *Journal of Network and Systems Management*, vol. 51, no. 7, p. 80, 2017.

[23] I. Pathak and D. P. Vidyarthi, "An optimal virtual network mapping model based on dynamic threshold," *Wireless Personal Communications*, vol. 83, no. 3, pp. 2381–2401, 2015.

[24] D. Jian, H. Tao, W. Jian, H. Wenbo, L. Jiang, and L. Yunjie, "Virtual network embedding through node connectivity," *The Journal of China Universities of Posts and Telecommunications*, vol. 22, no. 1, pp. 17–56, 2015.

[25] Q. Hu, Y. Wang, and X. Cao, Eds., *Towards Survivable Network Virtualization*. Piscataway, NJ and Piscataway, NJ: IEEE, 2013.

[26] R. Yu, C. Jiang, X. Xu, H. Cheng, and J. Wan, "Resources allocation in virtualized systems based on try- before-buy approach," *Procedia Environmental Sciences*, vol. 11, pp. 193–199, 2011.

[27] N. Triki, N. Kara, M. E. Barachi, and S. Hadjres, "A green energy-aware hybrid virtual network embedding approach," *Computer Networks*, vol. 91, pp. 712–737, 2015.

[28] E. Zola and A. J. Kassler, "Optimising for energy or robustness? trade-offs for vm consolidation in virtualized datacenters under uncertainty," *Optimization Letters*, vol. 11, no. 8, pp. 1571–1592, 2017.

[29] M. Amini Salehi, J. Smith, A. A. Maciejewski, H. J. Siegel, E. K. Chong, J. Apodaca, L. D. Briceño, T. Renner, V. Shestak, J. Ladd, A. Sutton, D. Janovy, S. Govindasamy, A. Alqudah, R. Dewri, and P. Prakash, "Stochastic-based robust dynamic resource allocation for independent tasks in a heterogeneous computing system," *Journal of Parallel and Distributed Computing*, vol. 97, pp. 96–111, 2016.

[30] B. LU, T. HUANG, X.-c. SUN, J.-y. CHEN, and Y.-j. LIU, "Dynamic recovery for survivable virtual network embedding," *The Journal of China Universities of Posts and Telecommunications*, vol. 21, no. 3, pp. 77–84, 2014.

[31] M. P. Anastasopoulos, A. Tzanakaki, and D. Simeonidou, "Stochastic planning of dependable virtual infrastructures over optical datacenter networks," *Journal of Optical Communications and Networking*, vol. 5, no. 9, p. 968, 2013.

[32] X. Chen, C. Li, and Y. Jiang, "A feedback control approach for energy efficient virtual network embedding," *Computer Communications*, vol. 80, pp. 16–32, 2016.

[33] Z. Chen, X. Zheng, and D. Lu, Eds., *A Study of a Virtual Network Embedding Algorithm*, 2015.

[34] M. Melo, S. Sargento, U. Killat, A. Timm-Giel, and J. Carapinha, "Optimal virtual network embedding: Node-link formulation," *IEEE Transactions on Network and Service Management*, vol. 10, no. 4, pp. 356–368, 2013.

[35] B. Naudts, M. Flores, R. Mijumbi, S. Verbrugge, J. Serrat, and D. Colle, "A dynamic pricing algorithm for a network of virtual resources," *International Journal of Network Management*, vol. 27, no. 2, p. e1960, 2017.

[36] X. Liu, Z. Zhang, X. Li, and S. Su, "Optimal virtual network embedding based on artificial bee colony," *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, no. 1, p. 756, 2016.

[37] M. Mangili, J. Elias, F. Martignon, and A. Capone, "Optimal planning of virtual content delivery networks under uncertain traffic demands," *Computer Networks*, vol. 106, pp. 186–195, 2016.

[38] X. Zhang, Z. Huang, C. Wu, Z. Li, and F. C. M. Lau, "Online stochastic buy-sell mechanism for vnf chains in the nfv market," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 2, pp. 392–406, 2017.

[39] N. Ogino, T. Kitahara, S. Arakawa, and M. Murata, "Virtual network embedding with multiple priority classes sharing substrate resources," *Computer Networks*, vol. 112, pp. 52–66, 2017.

[40] S. Cook, "The p versus np problem," in *Clay Mathematical Institute; The Millennium Prize Problem*, 2000.

[41] F. Rossi, P. van Beek, and T. Walsh, *Handbook of constraint programming*, ser. Foundations of artificial intelligence. Elsevier Science, 2006.

[42] "Ilog cplex: Optimization studio." [Online]. Available: https://www.ibm.com/products/ilog-cplex-optimization-studio?mhq=CPLEX&mhsrc=ibmsearch_p

[43] W. Szeto, Y. Iraqi, and R. Boutaba, "A multi-commodity flow based approach to virtual network resource allocation," vol. 3, pp. 1–18, 2003.

Appendix A

Case Study Information

In these appendices the information pertaining to node and links demands are provided for the case studies as well as financial information needed for comparative purposes.

A.1 Case Study Comparison Information

Table A.1: Case Study A Stochastic Improvement

Total User Cost	**Total WC Supplier Cost**	**Total WC Profit**	**Total SRA Supplier Cost**	**Total SRA Profit**
3390	2260	1130	1716	1674
Overall Stochastic Improvement 32.5%				

Table A.2: Case Study B Stochastic Improvement

Total User Cost	**Total WC Supplier Cost**	**Total WC Profit**	**Total SRA Supplier Cost**	**Total SRA Profit**
17310	10340	5170	9190	6860
Overall Stochastic Improvement 24.64%				

A.2 Case Study B Worst Case Demand Information

This information is concerned with the worst case version of case study B.

Req (R)	Node	Node Demand	Possible Profit
1	1	6	60
	2	5	50
	3	4	40
	4	5	50
	5	6	60
2	1	8	80
	4	10	100
	6	9	90
	7	6	60
3	6	8	80
	13	3	30
	19	7	70
	20	8	80
4	4	3	30
	5	3	30
	7	3	30
	8	3	30
	9	7	70
	10	6	60
	11	7	70
	12	8	80
5	13	6	60
	15	5	50
	20	6	60
6	8	4	40
	9	6	60
	10	4	40
	16	9	90
	17	9	90
	18	8	80
7	7	3	30
	8	3	30
	14	2	20
8	8	4	40
	13	4	40
	14	4	40
	15	6	60

	16	6	60
9	1	5	50
	4	3	30
	5	4	40
	6	4	40
	8	5	50
	13	4	40
	14	3	30
10	5	5	50
	8	5	50
	10	6	60
	11	6	60
	16	4	40
	17	3	30
	18	3	30
11	6	4	40
	7	3	30
	13	4	40
	14	4	40
	19	8	80
	20	6	60
12	8	4	40
	13	5	50
	14	3	30
	15	7	70
	16	5	50
	20	8	80
13	1	4	40
	2	7	70
	3	8	80
	4	3	30
	5	4	40
	6	3	30
	7	2	20
14	4	3	30
	5	5	50
	7	4	40
	8	5	50
	14	3	30

Table A.3: Node Request Demands Case Study B: Worst Case

Req (R)	Link Direction	Link Demand	Cost to Supplier	Cost to User	Possible Profit
1	(1,2)	15	300	450	150
	(2,3)	10	200	300	100
	(3,5)	7	140	210	70
	(5,4)	5	100	150	50
	(4,1)	6	120	180	60
2	(1,4)	12	240	360	120
	(4,7)	6	120	180	60
	(7,6)	12	240	360	120
	(6,1)	18	360	540	180
3	(6,19)	12	240	360	120
	(19,20)	10	200	300	100
	(20,13)	13	260	390	130
	(13,6)	12	240	360	120
4	(4,7)	7	140	210	70
	(7,8)	9	180	270	90
	(8,9)	10	200	300	100
	(9,10)	13	260	390	130
	(10,11)	10	200	300	100
	(11,12)	13	260	390	130
	(12,5)	15	300	450	150
	(5,4)	15	300	450	150
5	(13,20)	10	200	300	100
	(20,15)	12	240	360	120
	(15,13)	13	260	390	130
6	(8,9)	13	260	390	130
	(9,10)	14	280	420	140
	(10,18)	12	240	360	120
	(18,17)	11	220	330	110
	(17,16)	12	240	360	120
	(16,8)	13	260	390	130
7	(7,14)	9	180	270	90
	(14,8)	11	220	330	110
	(8,7)	8	160	240	80
8	(8,16)	11	220	330	110
	(16,15)	8	160	240	80
	(15,13)	9	180	270	90
	(13,14)	11	220	330	110
	(14,8)	11	220	330	110
9	(1,6)	7	140	210	70
	(6,13)	6	120	180	60
	(13,14)	7	140	210	70

	(14,8)	7	140	210	70
	(8,5)	8	160	240	80
	(5,4)	8	160	240	80
	(4,1)	8	160	240	80
10	(5,8)	13	260	390	130
	(8,16)	12	240	360	120
	(16,17)	12	240	360	120
	(17,18)	8	160	240	80
	(18,10)	11	220	330	110
	(10,11)	10	200	300	100
	(11,5)	12	240	360	120
11	(6,7)	9	180	270	90
	(7,14)	9	180	270	90
	(14,13)	7	140	210	70
	(13,20)	7	140	210	70
	(20,19)	8	160	240	80
	(19,6)	9	180	270	90
12	(8,14)	8	160	240	80
	(14,13)	8	160	240	80
	(13,20)	9	180	270	90
	(20,15)	9	180	270	90
	(15,16)	9	180	270	90
	(16,8)	8	160	240	80
13	(1,2)	11	220	330	110
	(2,3)	13	260	390	130
	(3,5)	12	240	360	120
	(5,4)	6	120	180	60
	(4,7)	13	260	390	130
	(7,6)	12	240	360	120
	(6,1)	11	220	330	110
14	(4,7)	10	200	300	100
	(7,14)	9	180	270	90
	(14,8)	6	120	180	60
	(8,5)	7	140	210	70
	(5,4)	7	140	210	70

Table A.4: Link Request Demands Case Study B: Worst Case

A.3 Case Study B Financial Information

These are financial results needed for comparative purposes, specifically concerning case study B.

Table A.5: Case Study B Financial Information Worst Case

Total User Cost													
R1	**R2**	**R3**	**R4**	**R5**	**R6**	**R7**	**R8**	**R9**	**R10**	**R11**	**R12**	**R13**	**R14**
2070	2430	1800	3960	1560	3450	1080	2220	2370	3300	2340	2490	3270	1770
Total Supplier Cost													
R1	**R2**	**R3**	**R4**	**R5**	**R6**	**R7**	**R8**	**R9**	**R10**	**R11**	**R12**	**R13**	**R14**
1380	1620	1200	2640	1040	2300	720	1480	1580	2200	1560	1660	2180	1180
Available Profit													
R1	**R2**	**R3**	**R4**	**R5**	**R6**	**R7**	**R8**	**R9**	**R10**	**R11**	**R12**	**R13**	**R14**
690	810	600	1320	520	1150	360	740	790	1100	780	830	1090	590

Table A.6: Case Study B Total User Cost and Total Supplier Cost SRA

	R1	**R2**	**R3**	**R4**	**R5**	**R6**	**R7**	**R8**	**R9**	**R10**	**R11**	**R12**	**R13**	**R14**
							Total User Cost							
S1	1860	2280	1860	3570	1440	3000	840	1950	2040	2730	1980	2250	2460	1410
S2	1800	1890	1830	3420	1410	2970	810	1980	2040	2580	1950	1740	2490	1530
S3	1680	2070	1530	3390	1320	2730	1020	1800	1890	2400	1920	1890	2700	1710
							Total Supplier Cost							
	R1	**R2**	**R3**	**R4**	**R5**	**R6**	**R7**	**R8**	**R9**	**R10**	**R11**	**R12**	**R13**	**R14**
S1	1240	1520	1240	2380	960	2000	560	1300	1360	1820	1320	1500	1640	940
S2	1200	1260	1220	2280	940	1980	540	1320	1360	1720	1300	1160	1660	1020
S3	1120	1380	1020	2260	880	1820	680	1200	1260	1600	1280	1260	1800	1140

A.4 Case Study B Stochastic Case Demand Information

This information is concerned with the stochastic version of case study B.

Req (R)	Sce (S)	Node	Node Demand	Cost to Supplier	Cost to User	Possible Profit
1	1	1	6	120	180	60
		2	5	100	150	50
		3	4	80	120	40
		4	5	100	150	50
		5	2	40	60	20
1	2	1	2	40	60	20
		2	4	80	120	40
		3	4	80	120	40
		4	5	100	150	50
		5	6	120	180	60
1	3	1	6	120	180	60
		2	1	20	30	10
		3	1	20	30	10
		4	5	100	150	50
		5	3	60	90	30
2	1	1	8	160	240	80
		4	10	200	300	100
		6	6	120	180	60
		7	6	120	180	60
2	2	1	8	160	240	80
		4	10	200	300	100
		6	7	140	210	70
		7	3	60	90	30
2	3	1	8	160	240	80
		4	6	120	180	60
		6	9	180	270	90
		7	2	40	60	20
3	1	6	8	160	240	80
		13	3	60	90	30
		19	7	140	210	70
		20	2	40	60	20
3	2	6	7	140	210	70
		13	2	40	60	20
		19	6	120	180	60
		20	8	160	240	80
3	3	6	5	100	150	50
		13	2	40	60	20

		19	2	40	60	20
		20	8	160	240	80
4	1	4	2	40	60	20
		5	3	60	90	30
		7	3	60	90	30
		8	2	40	60	20
		9	4	80	120	40
		10	4	80	120	40
		11	5	100	150	50
		12	8	160	240	80
4	2	4	3	60	90	30
		5	2	40	60	20
		7	3	60	90	30
		8	3	60	90	30
		9	7	140	210	70
		10	6	120	180	60
		11	4	80	120	40
		12	5	100	150	50
4	3	4	2	40	60	20
		5	3	60	90	30
		7	2	40	60	20
		8	3	60	90	30
		9	4	80	120	40
		10	5	100	150	50
		11	7	140	210	70
		12	8	160	240	80
5	1	13	6	120	180	60
		15	5	100	150	50
		20	4	80	120	40
5	2	13	4	80	120	40
		15	5	100	150	50
		20	6	120	180	60
5	3	13	2	40	60	20
		15	5	100	150	50
		20	4	80	120	40
6	1	8	2	40	60	20
		9	5	100	150	50
		10	4	80	120	40
		16	7	140	210	70
		17	7	140	210	70
		18	8	160	240	80
6	2	8	4	80	120	40
		9	6	120	180	60

		10	4	80	120	40
		16	7	140	210	70
		17	7	140	210	70
		18	8	160	240	80
6	3	8	3	60	90	30
		9	5	100	150	50
		10	4	80	120	40
		16	9	180	270	90
		17	9	180	270	90
		18	4	80	120	40
7	1	7	2	40	60	20
		8	3	60	90	30
		14	2	40	60	20
7	2	7	3	60	90	30
		8	2	40	60	20
		14	2	40	60	20
7	3	7	2	40	60	20
		8	2	40	60	20
		14	2	40	60	20
8	1	8	4	80	120	40
		13	3	60	90	30
		14	4	80	120	40
	15	5	100	150	50	
		16	6	120	180	60
8	2	8	2	40	60	20
		13	4	80	120	40
		14	2	40	60	20
		15	6	120	180	60
		16	4	80	120	40
8	3	8	3	60	90	30
		13	3	60	90	30
		14	2	40	60	20
		15	4	80	120	40
		16	6	120	180	60
9	1	1	4	80	120	40
		4	3	60	90	30
		5	4	80	120	40
		6	2	40	60	20
		8	5	100	150	50
		13	4	80	120	40
		14	2	40	60	20
9	2	1	5	100	150	50
		4	2	40	60	20

		5	3	60	90	30
		6	4	80	120	40
		8	2	40	60	20
		13	2	40	60	20
		14	3	60	90	30
9	3	1	2	40	60	20
		4	3	60	90	30
		5	4	80	120	40
		6	3	60	90	30
		8	2	40	60	20
		13	4	80	120	40
		14	2	40	60	20
10	1	5	4	80	120	40
		8	5	100	150	50
		10	6	120	180	60
		11	5	100	150	50
		16	4	80	120	40
		17	2	40	60	20
		18	3	60	90	30
10	2	5	5	100	150	50
		8	4	80	120	40
		10	5	100	150	50
		11	6	120	180	60
		16	2	40	60	20
		17	3	60	90	30
		18	2	40	60	20
10	3	5	3	60	90	30
		8	2	40	60	20
		10	2	40	60	20
		11	2	40	60	20
		16	4	80	120	40
		17	3	60	90	30
		18	2	40	60	20
11	1	6	2	40	60	20
		7	3	60	90	30
		13	3	60	90	30
		14	4	80	120	40
		19	7	140	210	70
		20	6	120	180	60
11	2	6	3	60	90	30
		7	2	40	60	20
		13	4	80	120	40
		14	2	40	60	20

		19	8	160	240	80
		20	6	120	180	60
11	3	6	4	80	120	40
		7	3	60	90	30
		13	2	40	60	20
		14	3	60	90	30
		19	6	120	180	60
		20	4	80	120	40
12	1	8	3	60	90	30
		13	5	100	150	50
		14	3	60	90	30
		15	7	140	210	70
		16	2	40	60	20
		20	8	160	240	80
12	2	8	4	80	120	40
		13	2	40	60	20
		14	3	60	90	30
		15	3	60	90	30
		16	5	100	150	50
		20	2	40	60	20
12	3	8	2	40	60	20
		13	5	100	150	50
		14	2	40	60	20
		15	4	80	120	40
		16	5	100	150	50
		20	2	40	60	20
13	1	1	2	40	60	20
		2	3	60	90	30
		3	4	80	120	40
		4	3	60	90	30
		5	4	80	120	40
		6	2	40	60	20
		7	2	40	60	20
13	2	1	4	80	120	40
		2	2	40	60	20
		3	6	120	180	60
		4	2	40	60	20
		5	3	60	90	30
		6	2	40	60	20
		7	2	40	60	20
13	3	1	2	40	60	20
		2	7	140	210	70
		3	8	160	240	80

		4	2	40	60	20
		5	4	80	120	40
		6	3	60	90	30
		7	2	40	60	20
14	1	4	2	40	60	20
		5	3	60	90	30
		7	3	60	90	30
		8	2	40	60	20
		14	2	40	60	20
14	2	4	3	60	90	30
		5	4	80	120	40
		7	2	40	60	20
		8	3	60	90	30
		14	2	40	60	20
14	3	4	3	60	90	30
		5	5	100	150	50
		7	4	80	120	40
		8	5	100	150	50
		14	3	60	90	30

Table A.7: Node Request Demands Case Study B: Stochastic

Req (R)	**Sce (S)**	**Link Direction**	**Link Demand**	**Cost to Supplier**	**Cost to User**	**Possible Profit**
1	1	(1,2)	15	300	450	150
		(2,3)	8	160	240	80
		(3,5)	7	140	210	70
		(5,4)	5	100	150	50
		(4,1)	5	100	150	50
1	2	(1,2)	13	260	390	130
		(2,3)	10	200	300	100
		(3,5)	6	120	180	60
		(5,4)	5	100	150	50
		(4,1)	5	100	150	50
1	3	(1,2)	12	240	360	120
		(2,3)	10	200	300	100
		(3,5)	7	140	210	70
		(5,4)	5	100	150	50
		(4,1)	6	120	180	60
2	1	(1,4)	12	240	360	120
		(4,7)	6	120	180	60
		(7,6)	10	200	300	100

		(6,1)	18	360	540	180
2	2	(1,4)	8	160	240	80
		(4,7)	5	100	150	50
		(7,6)	12	240	360	120
		(6,1)	10	200	300	100
2	3	(1,4)	12	240	360	120
		(4,7)	5	100	150	50
		(7,6)	12	240	360	120
		(6,1)	15	300	450	150
3	1	(6,19)	12	240	360	120
		(19,20)	10	200	300	100
		(20,13)	8	160	240	80
		(13,6)	12	240	360	120
3	2	(6,19)	11	220	330	110
		(19,20)	9	180	270	90
		(20,13)	10	200	300	100
		(13,6)	8	160	240	80
3	3	(6,19)	6	120	180	60
		(19,20)	7	140	210	70
		(20,13)	13	260	390	130
		(13,6)	8	160	240	80
4	1	(4,7)	7	140	210	70
		(7,8)	8	160	240	80
		(8,9)	10	200	300	100
		(9,10)	11	220	330	110
		(10,11)	9	180	270	90
		(11,12)	13	260	390	130
		(12,5)	15	300	450	150
		(5,4)	15	300	450	150
4	2	(4,7)	7	140	210	70
		(7,8)	8	160	240	80
		(8,9)	8	160	240	80
		(9,10)	13	260	390	130
		(10,11)	10	200	300	100
		(11,12)	8	160	240	80
		(12,5)	12	240	360	120
		(5,4)	15	300	450	150
4	3	(4,7)	7	140	210	70
		(7,8)	9	180	270	90
		(8,9)	8	160	240	80
		(9,10)	5	100	150	50
		(10,11)	10	200	300	100
		(11,12)	13	260	390	130

		(12,5)	15	300	450	150
		(5,4)	12	240	360	120
5	1	(13,20)	10	200	300	100
		(20,15)	12	240	360	120
		(15,13)	11	220	330	110
5	2	(13,20)	8	160	240	80
		(20,15)	11	220	330	110
		(15,13)	13	260	390	130
5	3	(13,20)	8	160	240	80
		(20,15)	12	240	360	120
		(15,13)	13	260	390	130
6	1	(8,9)	13	260	390	130
		(9,10)	14	280	420	140
		(10,18)	12	240	360	120
		(18,17)	11	220	330	110
		(17,16)	8	160	240	80
		(16,8)	9	180	270	90
6	2	(8,9)	8	160	240	80
		(9,10)	9	180	270	90
		(10,18)	10	200	300	100
		(18,17)	11	220	330	110
		(17,16)	12	240	360	120
		(16,8)	13	260	390	130
6	3	(8,9)	12	240	360	120
		(9,10)	12	240	360	120
		(10,18)	11	220	330	110
		(18,17)	9	180	270	90
		(17,16)	8	160	240	80
		(16,8)	5	100	150	50
7	1	(7,14)	8	160	240	80
		(14,8)	9	180	270	90
		(8,7)	4	80	120	40
7	2	(7,14)	5	100	150	50
		(14,8)	10	200	300	100
		(8,7)	5	100	150	50
7	3	(7,14)	9	180	270	90
		(14,8)	11	220	330	110
		(8,7)	8	160	240	80
8	1	(8,16)	8	160	240	80
		(16,15)	5	100	150	50
		(15,13)	9	180	270	90
		(13,14)	10	200	300	100
		(14,8)	11	220	330	110

8	2	(8,16)	11	220	330	110
		(16,15)	8	160	240	80
		(15,13)	9	180	270	90
		(13,14)	10	200	300	100
		(14,8)	10	200	300	100
8	3	(8,16)	6	120	180	60
		(16,15)	7	140	210	70
		(15,13)	8	160	240	80
		(13,14)	11	220	330	110
		(14,8)	10	200	300	100
9	1	(1,6)	5	100	150	50
		(6,13)	5	100	150	50
		(13,14)	6	120	180	60
		(14,8)	6	120	180	60
		(8,5)	7	140	210	70
		(5,4)	7	140	210	70
		(4,1)	8	160	240	80
9	2	(1,6)	6	120	180	60
	2	(6,13)	6	120	180	60
		(13,14)	7	140	210	70
		(14,8)	7	140	210	70
		(8,5)	8	160	240	80
		(5,4)	8	160	240	80
		(4,1)	5	100	150	50
9	3	(1,6)	7	140	210	70
		(6,13)	6	120	180	60
		(13,14)	5	100	150	50
		(14,8)	5	100	150	50
		(8,5)	7	140	210	70
		(5,4)	7	140	210	70
		(4,1)	6	120	180	60
10	1	(5,8)	11	220	330	110
		(8,16)	12	240	360	120
		(16,17)	10	200	300	100
		(17,18)	8	160	240	80
		(18,10)	9	180	270	90
		(10,11)	7	140	210	70
		(11,5)	5	100	150	50
10	2	(5,8)	5	100	150	50
		(8,16)	6	120	180	60
		(16,17)	7	140	210	70
		(17,18)	8	160	240	80
		(18,10)	11	220	330	110

		(10,11)	10	200	300	100
		(11,5)	12	240	360	120
10	3	(5,8)	13	260	390	130
	3	(8,16)	10	200	300	100
		(16,17)	12	240	360	120
		(17,18)	8	160	240	80
		(18,10)	5	100	150	50
		(10,11)	6	120	180	60
		(11,5)	8	160	240	80
11	1	(6,7)	8	160	240	80
		(7,14)	9	180	270	90
		(14,13)	7	140	210	70
		(13,20)	6	120	180	60
		(20,19)	5	100	150	50
		(19,6)	6	120	180	60
11	2	(6,7)	6	120	180	60
		(7,14)	5	100	150	50
		(14,13)	7	140	210	70
		(13,20)	5	100	150	50
		(20,19)	8	160	240	80
		(19,6)	9	180	270	90
11	3	(6,7)	9	180	270	90
		(7,14)	8	160	240	80
		(14,13)	5	100	150	50
		(13,20)	7	140	210	70
		(20,19)	6	120	180	60
		(19,6)	7	140	210	70
12	1	(8,14)	8	160	240	80
		(14,13)	8	160	240	80
		(13,20)	9	180	270	90
		(20,15)	9	180	270	90
		(15,16)	7	140	210	70
		(16,8)	6	120	180	60
12	2	(8,14)	7	140	210	70
		(14,13)	6	120	180	60
		(13,20)	7	140	210	70
		(20,15)	5	100	150	50
		(15,16)	8	160	240	80
		(16,8)	6	120	180	60
12	3	(8,14)	7	140	210	70
		(14,13)	5	100	150	50
		(13,20)	6	120	180	60
		(20,15)	8	160	240	80

		(15,16)	9	180	270	90
		(16,8)	8	160	240	80
13	1	(1,2)	8	160	240	80
		(2,3)	9	180	270	90
		(3,5)	5	100	150	50
		(5,4)	4	80	120	40
		(4,7)	13	260	390	130
		(7,6)	12	240	360	120
		(6,1)	11	220	330	110
13	2	(1,2)	11	220	330	110
		(2,3)	13	260	390	130
		(3,5)	12	240	360	120
		(5,4)	6	120	180	60
		(4,7)	5	100	150	50
		(7,6)	7	140	210	70
		(6,1)	8	160	240	80
13	3	(1,2)	8	160	240	80
		(2,3)	8	160	240	80
		(3,5)	6	120	180	60
		(5,4)	5	100	150	50
		(4,7)	13	260	390	130
		(7,6)	11	220	330	110
		(6,1)	11	220	330	110
14	1	(4,7)	8	160	240	80
		(7,14)	9	180	270	90
		(14,8)	6	120	180	60
		(8,5)	7	140	210	70
		(5,4)	5	100	150	50
14	2	(4,7)	10	200	300	100
		(7,14)	9	180	270	90
		(14,8)	6	120	180	60
		(8,5)	7	140	210	70
		(5,4)	5	100	150	50
14	3	(4,7)	10	200	300	100
		(7,14)	9	180	270	90
		(14,8)	6	120	180	60
		(8,5)	5	100	150	50
		(5,4)	7	140	210	70
		(5,4)	7	140	210	70

Table A.8: Link Request Demands Case Study B: Stochastic

A.5 Case Study B Comparison - Financial Information

These tables are used in the case study B comparison.

Table A.9: Case Study B Total User Cost Comparison

	Total User Cost													
	R1	**R2**	**R3**	**R4**	**R5**	**R6**	**R7**	**R8**	**R9**	**R10**	**R11**	**R12**	**R13**	**R14**
S1	1860	2280	1860	3570	1440	3000	840	1950	2040	2730	1980	2250	2460	1410
S2	1800	1890	1830	3420	1410	2970	810	1980	2040	2580	1950	1740	2490	1530
S3	1680	2070	1530	3390	1320	2730	1020	1800	1890	2400	1920	1890	2700	1710
MAX	2070	2430	1800	3960	1560	3450	1080	2220	2370	3300	2340	2490	3270	1770

Table A.10: Case Study B Total Supplier Cost Comparison

	Total Supplier Cost													
	R1	**R2**	**R3**	**R4**	**R5**	**R6**	**R7**	**R8**	**R9**	**R10**	**R11**	**R12**	**R13**	**R14**
S1	1240	1520	1240	2380	960	2000	560	1300	1360	1820	1320	1500	1640	940
S2	1200	1260	1220	2280	940	1980	540	1320	1360	1720	1300	1160	1660	1020
S3	1120	1380	1020	2260	880	1820	680	1200	1260	1600	1280	1260	1800	1140
MAX	1380	1620	1200	2640	1040	2300	720	1480	1580	2200	1560	1660	2180	1180

Appendix B

Conference Contributions

In this appendix the article submitted and presented at SATNAC is shown.

M. Hollestein, M.J. Grobler, S.E. Terblanche, *"Resource Allocation In Virtual Network Embedding"*, in Southern African Telecommunications and Network Access Conference (SATNAC), Hermanus, South Africa, September 2018

Resource Allocation In Virtual Network Embedding

M. Hollestein*, M.J. Grobler*, S.E. Terblanche †
*School of Electric,Electronic and Computer Engineering, North West University, South Africa
[1]marja.hollestein@gmail.com
[2]leenta.grobler@nwu.ac.za
†School of Industrial Engineering, North West University, South Africa
[3]fanie.terblanche@nwu.ac.za

***Abstract*—To prevent stagnation of internet infrastructure, Virtual Network Embedding (VNE) has emerged as a promising component of future internet. Even though virtual networks (VN's) are virtual, a substrate infrastructure is still needed to host them. Over-provisioning and non-optimal resource allocation are, therefore, still a critical consideration when implementing VNE. In this paper, a mixed integer linear program is used to model a VNE problem which includes off-line stochastic resource allocation (SRA). Using this approach the advantages of increased resource allocation is found to be financially beneficial for Internet Service Providers (ISP's). This paper highlights the gains that an ISP can obtain, that are preferable in a financial sense, as well as a positive impact on the environment by using less physical resources.**

***Index Terms*—resource allocation, stochastic optimization, stochastic resource allocation (SRA), virtual network embedding (VNE)**

I. Introduction

A network is a collection of nodes and links that are interconnected to form a larger system. Networks have a strong dependency on the hardware (physical equipment) needed to form the system, which means that a network is a very expensive and rigid infrastructure to operate. Implementing changes to the network can be difficult as it can be time-consuming, and in order to include any new functionality time and money is needed. All these factors cause the flexibility of a traditional network to be difficult [1].

Optimal resource allocation has become a focus in the technological environment due to the high costs associated with physical infrastructure and space. Furthermore, network operators have to keep up with the rate of change in this digital age [2]. This led to the creation of network virtualization. Virtualizing a network means that the functionality of the network becomes independent of the physical hardware that supports it. A major advantage is that the same physical server can be used for several different purposes - depending on the software installed - to enable faster and more efficient communication.

Hardware resources in virtual networks must be allocated carefully to ensure that a virtual network (VN) can operate at the same speed as traditional rigid networks, as well as achieve predictable and high levels of performance [3]. In a virtual network (VN) functions can be deployed on demand quickly and efficiently [4], thus from a financial perspective, virtual networks will be more efficient than traditional ones [1].

Internet Service Providers (ISPs) are currently viewed as two separate entities in order to provide dedicated networks and services separately. The first entity is an Infrastructure Provider (InP), who is responsible for the deployment, management and maintenance of network equipment. The second entity is known as a Service Provider (SP), who deploys network protocols and offers end-to-end services. An InP attends to different SPs and SPs can set up agreements with one or more InPs in order to use their physical resources [5].

In this paper, the effect of using stochastic elements is investigated when considering the resource allocation of virtual network embedding (VNE). In section II some background pertaining to VNE is provided. In section III related work is covered. Section IV and V consists of the model formulation and the computational results, respectively. Conclusions are drawn in section VI.

II. Virtual Network Embedding

Virtual network embedding (VNE) refers to the instance where multiple virtual networks are hosted on the same substrate network [1]. Substrate network simply refers to the underlying physical hardware [3]. Virtual network embedding is thus the instance where various network nodes are installed on the same infrastructure; this enables shared capacity between them. When a change in demand is observed it is not necessary to purchase new network hardware, it is simply a matter of changing the software to balance the capacity between nodes dynamically.

For multiple virtual networks to be hosted on the same substrate network at the same time, efficient resource allocation is required [1]. Each time a user needs to use their virtual network a request is made to the virtual embedded network system. This request is received and certain physical resources are allocated to this specific network request. The system allocates certain nodes and links to the request in order to accomplish the usage goal of the virtual network request.

In most instances, the allocation of resources in virtual networks is an inefficient and redundant process, which causes substrate resources to host less virtual networks than possible. Optimizing the allocation of virtual network resources is a notion that warrants further investigation.

III. Related Work

Feng et al. [5] considers high revenue and high acceptance ratios as their two major optimization objectives. This is because, from the perspective of InPs' it is preferential to embed as many VN requests as possible on the substrate network in order to maximize profit.

The problem of efficiently mapping virtual nodes and links of a VN request onto substrate network resources is known to be NP-hard [4–7] and consequently previous research works have focused on designing heuristic-based algorithms. These solutions have higher time complexity when considering the embedding process [8].

Jian et al. [9] concurs that several heuristic algorithms have been created, but argues that most of these solutions only consider local resources of nodes (such as CPU or bandwidth) and ignore the network attributes that can have a substantial impact on their embedding process. The authors formulated a model based on the attributes of the entire network and showed through extensive simulation that this improves the performance of VNE through a decrease in runtime and the improvement of acceptance ratio and revenue.

Chowdhury et al. [7] notes that the problem has mostly been approached as two sub-problems: the virtual node mapping (VNoM) and virtual link mapping (VLiM) problem; and has had a clear separation between these two phases of the problem. The authors of this paper argue that a collection of VN embedding algorithms that leverage a better coordination between these two phases would significantly increase the solution space and would improve the quality of embedding heuristics. Their proposed algorithm surpasses previous work in the field when considering different factors like acceptance ratio, revenue and provisioning cost. This shift in the field of research to coordinating the two subproblems has had a positive influence on the research field.

Robust resource allocation was addressed by Kosak et al. [10] by considering a self-organizing hierarchical system in coordination with an auction-based system. In this paper, uncertain demands are decreased by the anticipation of characteristic prediction errors and the development of schedules for the different possible outcomes of the demand. Chochlidakis et al. [8] also explores a robust optimization framework, but focusses on shortest path VNE where demands are uncertain rather than deterministic.

There has been some emerging work conducted on the energy impact when considering the VNE problem [11]. Melo et al. [12] proposed to solve the online VNE problem as an optimization problem with the main goal of minimizing the energy consumption for each VN mapping request. These energy aware approaches aim to do this while still optimizing the resource allocation of the VN requests [8].

Triki et al. [13] concur that more recently VNE approaches have focused on alternative objectives when considering their optimization. These objectives include energy efficiency and network security. This paper proposes a green energy-aware hybrid VNE that aimed to realize energy efficiency and resource integration while reducing the resulting CO2 emissions of VN operation.

Besiktas et al. [14] contends that some VN operators may not trust each other and would insist that their virtual infrastructure is not co-hosted on the same substrate equipment. This is an example of why security aware VNE is an active research field.

Gong et al. [2] states that new security risks are introduced by network virtualization. This paper addresses the security risks by incorporating security factors into the VN embedding process. This paper formulates a mixed integer linear program for the VNE problem for different levels of security risk modes and proposes heuristic algorithms to solve these models, aiming to enable more flexible VNE.

Research results on stochastic resource allocation when specifically considering VNE is minimal [10]. Fan et al. [15] indicates that the stochastic resource allocation (SRA) problem is a broad class of integrative optimization problems that are comprehensive in complex systems such as communication systems and unmanned systems. Where SRA differs from conventional resource allocation is that a certain probability is prescribed to the ability of a resource to complete a task and that the end objective is to maximize the payoff through the suitable assignment of an available resource to different tasks. Even though this paper does not focus on VNE it can still be valuable when considering stochastic optimization approaches.

The existing work in SRA in VNE is mostly in the online version of the problem [16], meaning the network request are known in advance. In this paper an offline learning approach is developed for batch training and is only then extrapolated to an online approach. This approach is therefore called a learn-and-adapt approach with the goal of online learning advances in order to facilitate online resource allocation tasks. The approach uses a form of offline allocation in order to obtain a sensible online solution. This is done to simplify the online problem and achieve fast convergence and tolerable delay.

Stochastic approaches are not widely available and approaches of the VNE differ in terms of approximation, substrate infrastructure, topology and many other factors.

IV. Model Formulation

Let the substrate infrastructure be represented by $G^s = (V^s, E^s)$, with node capacities m_i for all $i \in V^s$, and link capacities n_{ij} for all $(i,j) \in E^s$. Given a virtual network topology $G = (V, E)$, edge $(u, v) \in E$ has a capacity C_{uv}.

Let $R = \{1, ..., |R|\}$ be the set of VN requests. Each request $r \in R$ has a set of virtual nodes $V(r)$.

Stochastic scenarios are defined as the set $S = \{1, 2, ..., |S|\}$. Each of these scenarios consist of different demands for the same request. An example of this would be the difference in demand for a virtual network of a business during and after business hours.

There is a profit attached to each VN request given as p_{rs}, for each request $r \in R$ and its scenarios $s \in S$. This profit is calculated as the cost of hosting the VN subtracted from the earnings gained from the user of that specific VN. The costs of hosting a virtual network is seen as the cost of using nodes and links for a specific goal.

There are commodities defined by $K = \{1, 2, .., |K|\}$ that represents a commodity between two nodes and has a demand defined by d_k^{rs}, for commodity $k \in K$, request $r \in R$ and scenario $s \in S$.

Let $y^r \in \{0, 1\}$ receive the value of one if the request $r \in R$ is accepted or null otherwise. Let $x_{ui}^r \in \{0, 1\}$ be one if the virtual node $u \in V$ is placed on the substrate node $i \in V^s$ or otherwise null. Let $f_{ij}^{uv,rs} \in \{0, 1\}$ equal one if the flow of the virtual link $(u, v) \in E$ is placed on the substrate physical link $(i, j) \in E^s$ for a request $r \in R$ and scenario $s \in S$, otherwise null ensuring a link capacity is not exceeded.

The VNE problem can then be represented by the following MIP model:

$$\text{maximise} \sum_{s\in S}\sum_{r\in R} p_{rs} y^r \quad (1)$$

subject to the constraints

$$\sum_{i\in V^s} x_{ui}^r = y^r \qquad \forall r \in R, \forall u \in V \quad (2)$$

$$\sum_{r\in R}\sum_{k\in K}\sum_{u\in V} d_k^{rs} x_{ui}^r \leq m_i \qquad \forall i \in V^s, \forall s \in S \quad (3)$$

$$\sum_{r\in R}\sum_{k\in K} f_{ij}^{uv,rs} d_k^{rs} \leq C_{uv} \qquad \forall (i,j) \in E^s, \forall (u,v) \in E, \forall s \in S \quad (4)$$

$$\sum_{(i,j)\in E^s} f_{ij}^{uv,rs} = x_{ui}^r \qquad \forall r \in R, \forall u \in V, \forall (u,v) \in E, \forall s \in S \quad (5)$$

$$-\sum_{(j,i)\in E^s} f_{ij}^{uv,rs} = -x_{vj}^r \qquad \forall r \in R, \forall v \in V, \forall (u,v) \in E, \forall s \in S \quad (6)$$

$$\sum_{(i,j)\in E^s} f_{ij}^{uv,rs} - \sum_{(j,i)\in E^s} f_{ij}^{uv,rs} = 0 \qquad \forall r \in R, \forall (u,v) \in E, \forall s \in S \quad (7)$$

The objective function (1) needs to be maximized in order to obtain the highest possible profit. This means that the main goal is to place requests that will earn the most income for the supplier.

Constraint (2) ensures that every virtual node is only placed on a single physical node, which safeguards against the possibility of over- provisioning for the virtual nodes. This ensures that only one substrate node will be hosting the virtual node and resources are not wasted.

Constraint (3) and (4) are responsible for the node and link constraints separately, in order to ensure that the virtual node and links placed do not exceed the capacity of the substrate infrastructure.

Constraints (5) , (6) and (7) are the flow constraints of the model, these constraints ensure that the flow of the virtual network is achievable on the substrate infrastructure and does not interfere with the other networks that may be present in the system.

Constraint (5) ensures the flow at the source node is correct and that a flow must exit its source node completely. Constraint (6) ensures that a flow must enter its destination node completely. Constraint (7) ensures that the flow entering a node is equal to the flow that exits that specific node.

V. Computational Results

A. Performance Metrics

In order to compare the different results obtained in this experiment, certain performance measures must be in place in order to demonstrate differences in value. The performance measure that will be used in this experiment is the financial gain that can be achieved through the placement of virtual network requests. Other performance measures are discussed in the future work section VII.

There are different cost perspectives associated with a VNE implementation. Firstly there is the cost from the supplier side, this is the cost to host a virtual network from the perspective of node and link capacity. When a request is received the sensible course, for the supplier, is to only host the minimum node and link resources needed for a specific request, since this allows the supplier to save in costs through not over-provisioning a specific request and allow the other resources in the virtual network to be used for another virtual network request, if needed.

On the other hand, there is the cost associated with the user of the virtual network. The user pays a certain cost which links the size and demand of the user's requests to a financial equivalent.

The profit associated with each request is dependant on the maximum demand of that specific request, since the profit of a request is the supplier cost subtracted from the financial gain received from the user. Thus the profit margin can be increased if the supplier cost is decreased for certain periods of time while the maximum user cost remains consistent.

B. Experimental Set-up

In this article an off-line embedding approach is followed. This entails that requests are not embedded continuously every time a new request is received, but that all requests received are logged so that at certain time intervals (every hour for example) the virtual network requests can be processed.

The substrate network used for this article is a simple infrastructure and the resulting virtual network available is a simple structure. This is to enable the easy interpretation of the financial gain stimulated by the use of stochastic elements.

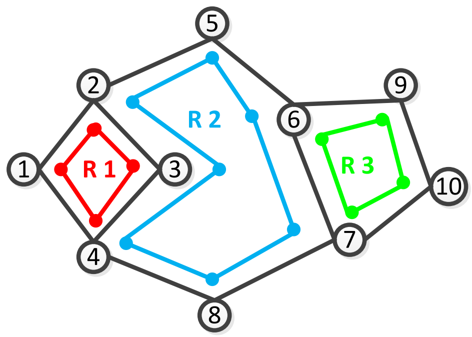

Fig. 1. Basic Structure with Requests

Refer to figure 1 for the substrate infrastructure of the available virtual network. This figure shows the available nodes and links in the virtual network (black). Figure 1 also shows the VN requests that need to be placed. There are three separate requests which are labelled as request 1(R1), request 2(R2) and request 3(R3). The information pertaining to these requests are shown later in this section.

Each user's request has certain scenarios coupled with that specific request to enable stochastic resource allocation (SRA). In this experimental set-up, there will be two scenarios for each request. One scenario will be a high demand scenario, which will emulate a large requirement of the virtual network. The second scenario is considered a low demand scenario, this will imitate - for example - business hours during which a business is not highly active.

The real world data for these two scenarios will be compiled using historical data of certain requests made by specific users. In reality, it cannot be assumed that one scenario will be a maximum scenario since another scenario can have certain demands that are higher than the scenario perceived as the maximum scenario, this is the reason why historical data plays an important role when considering real-world applications.

This will be compared to an experimental set-up of the same nature, but which has no scenarios in order to compare the financial gain achieved over existing methods that do not use SRA approaches.

Each request has a certain financial gain associated with that specific request, this financial gain is also coupled with the different scenarios for which the request can be placed. From the perspective of the supplier of the virtual network (an ISP for example), the financial gain is calculated by the cost per bandwidth that the user will pay during a user's request. A certain profit is also linked to accepting a network request, which varies in accordance with the urgency of the request and the number of resources needed.

For this case study the cost per capacity of the supplier of the infrastructure will be simplified, as well as the cost of the user per demand. This is done in order to easily show the effect of stochastic resource allocation on the financial gain in virtual network embedding. These values are shown in table I.

The node and link capacities of this structure are shown

TABLE I
FINANCIAL INFORMATION

Function	Cost to User	Cost to Supplier
Nodes	30	20
Links	30	20

respectively in Table II and Table III. These tables also indicate the cost of the supplier of the network per node or link, as well as the cost of the end user when that capacity is allocated to that user, which is used later in this article.

TABLE II
NODE CAPACITIES

Node	Capacity	Maximum Cost to Supplier	Maximum Cost to User	Maximum Profit
1	5	100	150	50
2	6	120	180	60
3	7	140	210	70
4	8	160	240	80
5	5	100	150	50
6	6	120	180	60
7	7	140	210	70
8	9	180	270	90
9	5	100	150	50
10	6	120	180	60
		Total Profit		640

TABLE III
LINK CAPACITIES

Link Pair	Total Capacity	Directional Capacity	Maximum Cost to Supplier	Maximum Cost to User	Maximum Profit
(1,2)	20	10	200	300	100
(1,4)	28	14	280	420	140
(2,3)	24	12	240	360	120
(3,4)	30	15	300	450	150
(2,5)	28	14	280	420	140
(4,8)	24	12	240	360	120
(5,6)	30	15	300	450	150
(8,7)	26	13	260	390	130
(6,7)	20	10	200	300	100
(6,9)	24	12	240	360	120
(7,10)	24	12	240	360	120
(9,10)	26	13	260	390	130

The network requests for the different scenarios are shown in Table IV. This table describes the demand for each request for each scenario. This table also shows the cost incurred by the supplier of the infrastructure for each request, as well as each scenario for that request. For comparative purposes, it is assumed that for each request the first scenario (scenario 1) is actively used 60% of the time that the VN is embedded and the second scenario (scenario 2) is used for the remaining 40% of the VN embedding time. This is referred to as the 60/40 division later in the article.

A summary of the financial information can be seen in table VII. The total demand for each request and its resulting scenarios are given in subdivisions, in terms of the node and link demands. The cost to the supplier is given, this is the cost to host the request for that specific scenario determined by the capacity needed for that request. The cost to the user is also given, which is the cost incurred by the user of the VN, determined by the demand of a specific request. The

TABLE IV
REQUEST DEMANDS

Req (R)	Scen (S)	Link Direction	Link Demand	Cost to Supplier	Cost to User	Possible Profit
1	1	(1,2)	9	180	270	90
		(2,3)	10	200	300	100
		(3,4)	8	160	240	80
		(4,1)	11	220	330	110
1	2	(1,2)	3	60	90	30
		(2,3)	3	60	90	30
		(3,4)	4	80	120	40
		(4,1)	5	100	150	50
2	1	(2,5)	10	200	300	100
		(5,6)	12	240	360	120
		(6,7)	8	160	240	80
		(7,8)	11	220	330	110
		(8,4)	10	200	300	100
		(4,3)	12	240	360	120
		(3,2)	10	200	300	100
2	2	(2,5)	6	120	180	60
		(5,6)	6	120	180	60
		(6,7)	5	100	150	50
		(7,8)	2	40	60	20
		(8,4)	2	40	60	20
		(4,3)	7	140	210	70
		(3,2)	4	80	120	40
3	1	(6,9)	10	200	300	100
		(9,10)	12	240	360	120
		(10,7)	9	180	270	90
		(7,6)	8	160	240	80
3	2	(6,9)	7	140	210	70
		(9,10)	2	40	60	20
		(10,7)	3	60	90	30
		(7,6)	5	100	150	50

user cost is calculated using the maximum demand for that user's request. This is how users are charged for their VN, by the maximum demand, non-inclusive of the scenario. The profit of a certain request can thus be calculated through the subtraction of the implementation cost of the supplier from the capital gained from the user.

C. Simulation and Computational Results

Implementing the aforementioned model using the structure and data given in the above section yields the following results shown in table V. This means that request 1 (R1) and request 3 (R3) have been placed in this specific simulation.

TABLE V
SIMULATION RESULTS

Request	Y
1	1
2	0
3	1

The results of the simulation are visualized in figure 2 where the request placement and directions are shown.

With the profit shown in table VI by comparing the cost of the supplier of the virtual network infrastructure and other resources to the cost incurred by the end user of the virtual network. This table firstly shows the profits of the model when no SRA is taken into consideration, thus the maximum demand is placed for each request, not considering what demand is actually being used by the user. The supplier cost

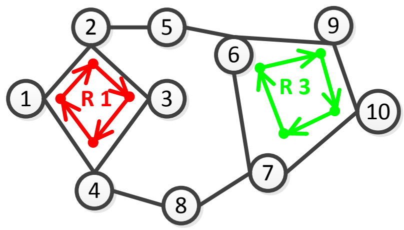

Fig. 2. VN Structure with Request Placements Results

is also taken as the maximum capacity allocated to a user request, this is thus a waste of infrastructure on the supplier's part. This is what is widely used in existing models and what is used in this article as a baseline comparison for this SRA method.

The second part of this table shows the costs associated when considering SRA using the 60/40 division discussed earlier in this article. The third part of this table compares the total profit that is available to the supplier of the VN. As it can be seen an average of 28.87% savings are obtained. In figure 3 the financial gain comparison is visualized. Each request is shown with the user cost (the income that is gained from the client), the supplier cost (the cost to host that VN request for the supplier) and the profit that the supplier can make. The income from the user does not change, but the profit and supplier cost is shown for non-stochastic allocation as well as stochastic allocation, which shows the increase in profit for the supplier possible from the use of SRA.

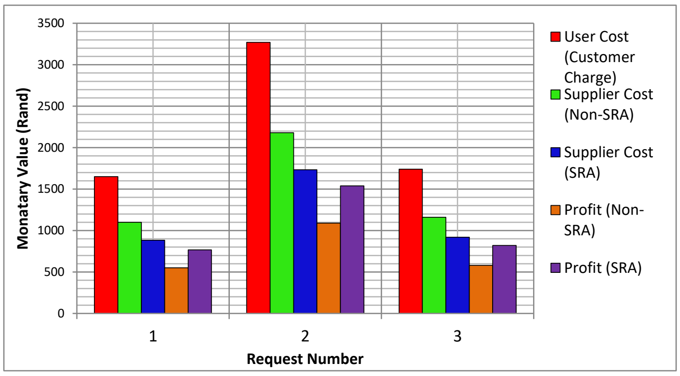

Fig. 3. Financial Summary

VI. CONCLUSION

When considering the effect that stochastic resource allocation can have over a deterministic modelling approach the conclusion can be made that the use of stochastic optimization techniques can increase the profits of ISP as well as the experience of the end users. It is shown that an average of 28.87% savings can be obtained simply through the use of stochastic resource allocation. This increase is from the use of SRA, which is not used in most existing works. This paper only has two scenarios which reflect a high and low demand, but in reality, different scenarios will not be classified into these two categories. Many scenarios can be available for a single request with node and link demands that differ in range.

TABLE VI
PROFIT

Request	Non-Stochastic Profit			Stochastic Profit				Comparison
	User Cost	Supplier Cost	Profit Normal	Scenario	Usage %	Supplier Cost	Profit	Difference Stochastic
1	-	-	-	1	60	660	-	-
	-	-	-	2	40	224	-	-
	1650	1100	550	*Total*	-	884	766	28.2%
2	-	-	-	1	60	1308	-	-
	-	-	-	2	40	424	-	-
	3270	2180	1090	*Total*	-	1732	1538	29.13%
3	-	-	-	1	60	696	-	-
	-	-	-	2	40	224	-	-
	1740	1160	580	*Total*	-	920	820	29.27%

TABLE VII
FINANCIAL SUMMARY

NODE DEMAND									
	Request 1	Supplier Cost	User Cost	Request 2	Supplier Cost	User Cost	Request 3	Supplier Cost	User Cost
Scenario 1	17	340	510	36	720	1080	19	380	570
Scenario 2	13	260	-	21	420	-	110	220	-
LINK DEMAND									
	Request 1	Supplier Cost	User Cost	Request 2	Supplier Cost	User Cost	Request 3	Supplier Cost	User Cost
Scenario 1	38	760	1140	73	1460	2190	39	780	1170
Scenario 2	15	300	-	32	640	-	17	340	-

The advantage that this scenario-based approach has is that during the times when the scenario used by a certain request allows it, another request can be placed on the same infrastructure which would not have been possible if the maximum resources were allocated to the original request.

Over-provisioning can also be reduced through the use of optimization techniques which in turn reduces the physical infrastructure needed to host the same amount of virtual networks. Considering the fact that physical space is a finite resource in our ever-expanding world this is an advantage that can positively influence this situation and create less waste in the physical environment. This is advantageous to the environment.

VII. FUTURE WORK

Certain performance measures were not explored in this article, namely the acceptance ratio, the embedding time, network security and energy efficiency of the embedding. If the scale of the experiment is widened, these factors can be significantly tested to further the influence this research can have in the field.

REFERENCES

[1] A. Fischer, J. F. Botero, M. T. Beck, H. de Meer, and X. Hesselbach, “Virtual network embedding: A survey,” *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 1888–1906, 2013.

[2] C. Besiktas, D. Gozupek, A. Ulas, E. Lokman, O. Ozyurt, and K. Ulusoy, “Flexible bandwidth-based virtual network embedding,” pp. 612–617, 2016.

[3] H. T. Nguyen, A. V. Vu, D. L. Nguyen, H. van Nguyen, M. N. Tran, Q. T. Ngo, T.-H. Truong, T. H. Nguyen, and T. Magedanz, “A generalized resource allocation framework in support of multi-layer virtual network embedding based on sdn,” *Computer Networks*, vol. 92, pp. 251–269, 2015.

[4] E. Amaldi, S. Coniglio, A. M. Koster, and M. Tieves, “On the computational complexity of the virtual network embedding problem,” *Electronic Notes in Discrete Mathematics*, vol. 52, pp. 213–220, 2016.

[5] M. Feng, J. Liao, S. Qing, T. Li, and J. Wang, “Cove: Co-operative virtual network embedding for network virtualization,” *Journal of Network and Systems Management*, vol. 51, no. 7, p. 80, 2017.

[6] Z. Chen, X. Zheng, and D. Lu, “A study of a virtual network embedding algorithm,” in *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, pp. 814–818.

[7] M. Chowdhury, M. R. Rahman, and R. Boutaba, “Vineyard: Virtual network embedding algorithms with coordinated node and link mapping,” *IEEE/ACM Transactions on Networking*, vol. 20, no. 1, pp. 206–219, 2012.

[8] G. Chochlidakis and V. Friderikos, “Robust virtual network embedding for mobile networks: Ieee pimrc 2015,” in *2015 IEEE 26th International Symposium on Personal, Indoor and Mobile Radio Communications - (PIMRC): Mobile and Wireless Networks*, pp. 1867–1871.

[9] D. Jian, H. Tao, W. Jian, H. Wenbo, L. Jiang, and L. Yunjie, “Virtual network embedding through node connectivity,” *The Journal of China Universities of Posts and Telecommunications*, vol. 22, no. 1, pp. 17–56, 2015.

[10] O. Kosak, G. Anders, F. Siefert, and W. Rief, “An approach to robust resource allocation in large-scale systems of systems,” in *2015 IEEE 9th International Conference on Self-Adaptive and Self-Organizing Systems*.

[11] X. Chen, C. Li, and Y. Jiang, “A feedback control approach for energy efficient virtual network embedding,” *Computer Communications*, vol. 80, pp. 16–32, 2016.

[12] M. Melo, S. Sargento, U. Killat, A. Timm-Giel, and J. Carapinha, “Optimal virtual network embedding: Energy aware formulation,” *Computer Networks*, vol. 91, pp. 184–195, 2015.

[13] N. Triki, N. Kara, M. E. Barachi, and S. Hadjres, “A green energy-aware hybrid virtual network embedding approach,” *Computer Networks*, vol. 91, pp. 712–737, 2015.

[14] C. Beşiktaş, D. Gözüpek, A. Ulaş, and E. Lokman, “Secure virtual network embedding with flexible bandwidth-based revenue maximization,” *Computer Networks*, vol. 121, pp. 89–99, 2017.

[15] G. Fan and H. Huang, “Scenario-based stochastic resource allocation with uncertain probability parameters,” *Journal of Systems Science and Complexity*, vol. 30, no. 2, pp. 357–377, 2017.

[16] T. Chen, A. Mokhtari, X. Wang, A. Ribeiro, and G. B. Giannakis, “Stochastic averaging for constrained optimization with application to online resource allocation,” *IEEE Transactions on Signal Processing*, vol. 65, no. 12, pp. 3078–3093, 2017.

Marja Hollestein is persuing her MEng degree in Computer/Electronic Engineering. She received her BEng in Electric/Electronic Engineering in 2016 and is planning to specialize in optimization and process control.