Spring 5-11-2015

Towards a Virtualized Next Generation Internet

Qian Hu

TOWARDS A VIRTUALIZED NEXT GENERATION INTERNET

by

QIAN HU

Under the Direction of Xiaojun Cao, PhD

ABSTRACT

A promising solution to overcome the Internet ossification is network virtualization in which Internet Service Providers (ISPs) are decoupled into two tiers: service providers (SPs), and infrastructure providers (InPs). The former maintain and customize virtual network(s) to meet the service requirement of end-users, which is mapped to the physical network infrastructure that is managed and deployed by the latter via the Virtual Network Embedding (VNE) process. VNE consists of two major components: node assignment, and link mapping, which can be shown to be NP-Complete.

In the first part of the dissertation, we present a path-based ILP model for the VNE problem. Our solution employs a branch-and-bound framework to resolve the integrity constraints, while embedding the column generation process to effectively obtain the lower bound for branch pruning. Different from existing approaches, the proposed solution can either obtain an optimal solution or a near-optimal solution with guarantee on the solution quality.

A common strategy in VNE algorithm design is to decompose the problem into two sequential sub-problems: node assignment (NA) and link mapping (LM). With this approach, it is inexorable to sacrifice the solution quality since the NA is not holistic and not-reversible. In the second part, we are motivated to answer the question: Is it possible to maintain the simplicity of the Divide-and-Conquer strategy while still achieving optimality? Our answer is based on a decomposition framework supported by the Primal-Dual analysis of the path-based ILP model.

This dissertation also attempts to address issues in two frontiers of network virtualization: survivability, and integration of optical substrate. In the third part, we address the survivable network embedding (SNE) problem from a network flow perspective, considering both splittable and non-splittable flows. In addition, the explosive growth of the Internet traffic calls for the support of a bandwidth-abundant optical substrate, despite the extra dimensions of complexity caused by the heterogeneities of optical resources, and the physical feature of optical transmission. In this fourth part, we present a holistic view of motivation, architecture, and challenges on the way towards a virtualized optical substrate that supports network virtualization.

INDEX WORDS: Network Virtualization, Virtual Network Embedding, Network Survivability, Optical Virtualization

TOWARDS A VIRTUALIZED NEXT GENERATION INTERNET

by

QIAN HU

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

in the College of Arts and Sciences

Georgia State University

2015

TOWARDS A VIRTUALIZED NEXT GENERATION INTERNET

by

QIAN HU

ACKNOWLEDGEMENTS

I have received encouragement, support and help from many great people during the past several years. I would never have been able to finish my dissertation without them.

I would like to express my gratitude to my advisor, Dr. Xiaojun Cao, for his support, help and encouragement through my Ph.D. study. He is open-mind, and always trusts and encourages me to explore new topics independently.

I would like to thank Dr. Guantao Chen for his guide as a mentor and care as a friend. I learned a lot from his probability class and received valuable suggestions on my research.

I would like to thank Dr. Anu Bourgeois for her great kindness and caring during my hard time. She is not only a great professor but also the nicest person that I have ever met.

I owe many thanks to Dr. Raj Sunderraman for his consistent help, encouragement, care, and guidance in teaching. He is always there to help me, like a father.

Many thanks to my friends: Huaping Ding, Songling Shan, Kebina & Ayush, Barnali, and Dr. Chaoyang Li, for their kindness whenever I need help.

Finally, I want to thank my family: my parents, my husband and my son. They are the greatest spiritual support of mine.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

PART 1

INTRODUCTION

The last decade has witnessed the stagnation of the current Internet. Due to the inherence resistance to technical advancement, revolutionary technologies can hardly be accepted. For instance, IPv6, conceived back in 1998, possesses only 1% share of the Internet traffic until late November 2012. A promising solution to overcome this ossification is network virtualization [1] [2] [3] [4] [5]. With network virtualization, the traditional Internet Service Providers (ISPs) are decoupled into two tiers: the service providers (SPs), and the infrastructure providers (InPs). The former maintain and customize virtual network(s) to meet the service requirement of end users, while the latter deploy and manage the physical network infrastructure that instantiates the virtual network request from the former. This decoupling provides the SPs a virtualized view of the underlying network infrastructure as well as the architecture-oblivious freedom of adopting revolutionary technologies [1] [4] [5]. Likewise, the InPs can transparently advance the physical network without service disruptions rippled to the SPs.

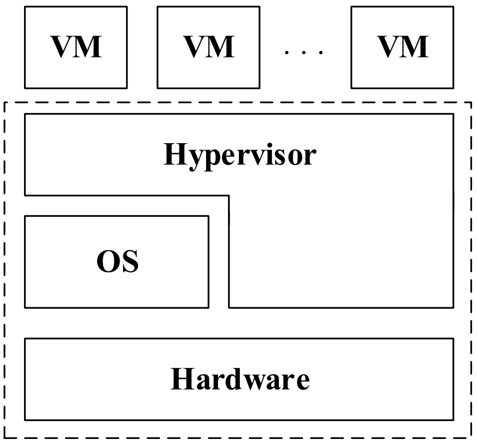


Figure 1.1. Server Virtualization

From the technological viewpoint, network virtualization reaps the advancement from both the node virtualization and link virtualization to support the abstraction of networking resource usage as an integrated virtual or logical network. Node virtualization, also known as server virtualization shown in 1.1, employs hypervisor (either upon the OS or bare metal hardware) to abstract and share the substrate software among multiple virtual machine (VM) instances. Network virtualization pushed this idea further to support a virtualized network that consists of virtual machines and the data connectivity in-between, as shown in Fig. 1.2. A list of enabling technologies are presented in Table 1.1. Given the functional difference, we separate the nodes into facility nodes (which mainly provide utility functions such as computing and storages) and switch nodes (which mainly support the traffic routing/switching). The facility node virtualization relies on the virtualization of OS and network interface card (NIC)(e.g., *Xen* [6]), and virtualization of switch nodes replies on the virtualization of routing functionalities. At the link level, a virtual data path can be created with the support of bandwidth multiplexing and technologies such as label/flow-based switching (e.g., *OpenFlow* [7]).

Table 1.1. Enabling Technologies of Network Virtualization

Component		**Enabling Technologies**	**Examples of Implementation**
Node	Facility node	OS, NIC Virtualization	Xen, VMware
	Switch node	Routing function virtualization	Router in virtual OS
Link		Bandwidth multiplexing, Lable/Flow Switching, Tunneling	Open Flow, MPLS

With the support of above virtualization technologies, a virtual network can be customized by an SP and mapped to the substrate network of the InPs via a process, known as *Virtual Network Embedding* (VNE). This process consists of two major components: node assignment, which is the mapping of the virtual node (with computational capacity requirement) to the substrate node; and link mapping, which is the mapping of the virtual link (with bandwidth capacity requirement) to the substrate path(s). Given the NP-Completeness of the VNE problem [8], existing approaches can be broadly classified into a few categories: optimal solutions based on solving the link-based Integer Linear Programming (ILP) formulation of virtual network embedding (e.g., [1]); approaches based on the relaxation of the ILP formulation (e.g., relaxation and rounding in [9]); and heuristic/meta-heuristic algorithms (e.g., [10]). The VNE process can be transformed into a classic multi-commodity flow problem [9] [11]. This leads to two variations of the virtual link mapping: splittable flow mapping and non-splittable flow mapping [12]. In general, the optimal solution based on ILP models suffers from the extensive computational time of the ILP solver in practice, while the relaxation or heuristics cannot provide a near-optimal solution with guarantee on closeness to the optimality.

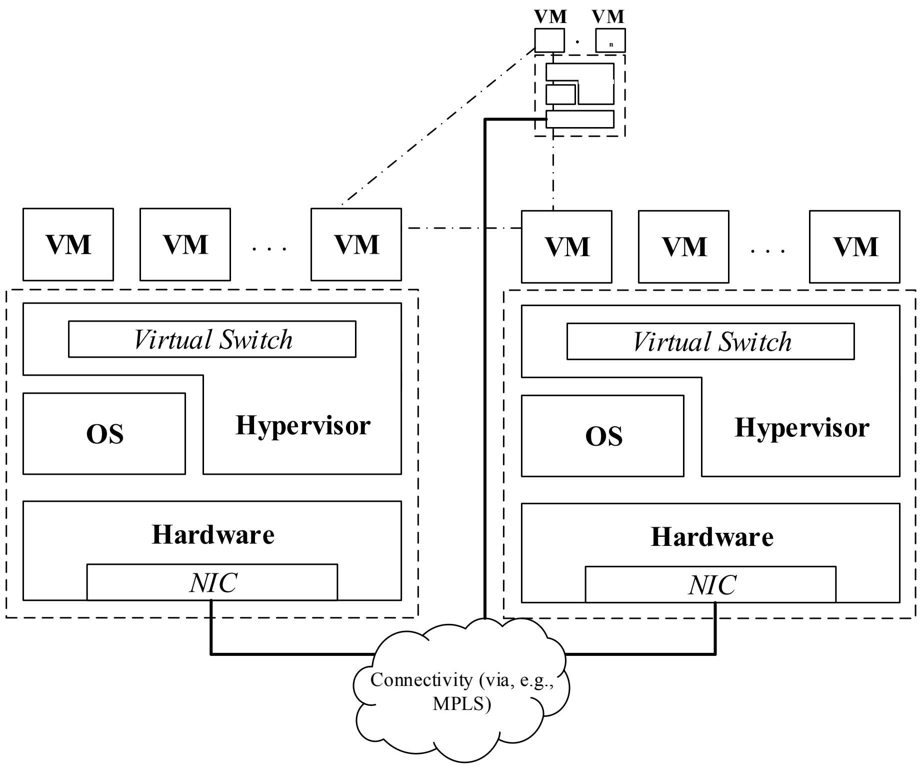


Figure 1.2. Network Virtualization

The first part of this dissertation aims to fill these gaps with a novel *branch and bound* framework for the VNE problem. Specifically, we present a compact path-based ILP model for the VNE problem. Our solution employ a *branch and bound* framework to resolve the integrity constraints while embedding a novel *column generation* process to effectively obtain the lower bound for branch pruning. Different from existing approaches, the proposed framework can either obtain an optimal solution or a near-optimal solution with guarantee on the solution quality.

Given the nature of the VNE problem, a simple and straightforward strategy for design VNE algorithms (i.e., relaxation and heuristics) is to decompose the problem into two sequential subproblems: node assignment (NA) and link mapping (LM), where the node assignment (NA) is decided first and then the link mapping problem is resolved under a fixed node mapping. With this approach, it is inexorable, however, to sacrifice the solution quality since the node assignment decision is not holistic and not-reversible. We hence are motivated by the following question: Is it possible to maintain the *simplicity* of the *Divide-and-Conquer* strategy while still achieving *optimality* for the VNE problem? The second major piece presented in this dissertation answers this question with an intelligent decomposition framework supported by the *Primal-Dual* analysis of the path-based ILP model that is proposed in the first part.

With network virtualization gaining momentum, the concept of survivable network virtualization attracts considerable attention recently. The resulted problem, namely *survivable virtual network embedding* (SNE) generally has to deal with failures of variety of network elements. In the literature, various types of failure scenarios, including single link, single facility node and single regional failure, have been investigated [13] [14] [15] [16] [17]. Different from the literature, we address the *survivable network embedding (SNE)* problem from a network flow perspective under two different cases: the case with non-splitable network flows and splitable network flows. In the former case, a virtual link can only be mapped to one substrate path while the latter cases allows the mapping of one virtual link to multiple substrate paths. Our extensively evaluation also reviews the tradeoff in terms of QoS and resource usages between these two cases.

Finally, the explosive growth of the Internet traffic clearly calls for the support of a bandwidth-abundant and energy-efficient optical substrate [18] [19]. Ideally, a virtualized optical substrate can provide any-to-any bandwidth-abundant connectivity for the service layer in network virtualization in an elastic, agile and automated manner with effective abstraction, partition/aggregation of optical resources. However, the heterogeneities of optical resources, and the analog feature of optical transmission add extra dimensions of complexity to the VNE problem. In this dissertation, we present a holistic view of motivation and architecture, and explicitly assess challenges on the way towards a virtualized optical substrate that supports network virtualization.

Overall, this dissertation addresses key challenges in network virtualization, and can serve as a basis for enabling a virtualized future Internet when combined together. The rest of this dissertation is organized as follows. In Part 2, we present important concepts, architectures and key enabling technologies in network virtualization. Related literature studies on VNE and SNE problems are comprehensively reviewed and discussed. Part 3 presents the branch and bound framework for VNE based on a path-based ILP model. In Part 4, an optimal framework based on the Divide and Conquer or decomposition strategy is presented for the VNE problem. We present our recent research results for the SNE problem in Part 5. In Part 6, optical-based network virtualization is discussed and studied. Finally, in Part 7, we conclude this paper, and discuss a few open problems in network virtualization that will possible be our focus of future work.

PART 2

NETWORK VIRTUALIZATION: BASIC CONCEPTS

In this chapter, we introduce the basic concept of network virtualization, and discuss the motivation for a virtualized future Internet. We also review related work in virtual network embedding, survivable network embedding, and optical-based network virtualization.

2.1 What is Network Virtualization?

The IT industry is experiencing an era of virtualization where software, platform, infrastructure, (or anything), are all abstracted and virtualized as services, enabling a *pay-as-you-go* business model. The major driver of this revolution is the resulted cost savings, management overhead reduction, as well as increased adaptability. Built upon these technologies, network virtualization reaps the advancement from both the node virtualization (e.g., *Xen* [6]) and link virtualization (e.g., *OpenFlow* [7]) to support the abstraction of networking resource usage as an integrated virtual or logical network.

In the literature, various definitions of Network Virtualization from different perspectives have been given. In this dissertation, we adapt a comparatively generic definition based on [20].

Definition: **Network Virtualization** is any form of partition or aggregation on a group of network resources, through which each user has a unique, separate view of the network. Particularly, network resources can be fundamental (nodes, links) or derived (topologies), which can be virtualized recursively.

From a business perspective, network virtualization entails a new model that decouples the traditional Internet service providers (ISPs) into two relatively independent roles: the service providers (SPs) and the infrastructure providers (InPs). The SPs lease resources from one or multiple InPs to provide end-to-end services to users (including common customers and/or other service providers) by programming allocated network resources to create/deploy virtual networks. The InPs possess and manage the underlying physical network resources, and provide resources to different services providers through programmable interfaces.

A deeper understanding of network virtualization can be realized through two important concepts: recursion and revisitation. First, a virtual network can be provided as service/resource to another virtual network, which is know as *recursion* or nesting. The features of the virtual network (parent virtual network) provided to another virtual network (child virtual network) can be inherited by its descendent. Second, *revisitation* happens when multiple virtual nodes of a single virtual network are allowed to be hosted by the same physical node. In a network virtualization environment, multiple virtual networks (requests/services) from different or the same SPs can co-exist. Figure 2.1 [1] shows an example of a network virtualization environment. In this example, two virtual networks VN1 and VN2 created by service provider SP1 and SP2 are two independent services to the users U1, U2, and U3. This is achieved by utilizing the physical network resources from the substrate network provided by InP1 and InP2 through partition or aggregation. Note that recursion happens between VN1 and VN2, thereinto, VN1 is the parent virtual network, while VN2 is the child. Revisitation exists since two virtual nodes in VN2 are mapped to the same substrate node provided by InP1.

Finally, as network virtualization creates an extra layer of virtual network. It is a fundamental problem that how to map and instantiate the virtual network to the substrate network while respecting the physical resource limitations. This problem is known as the *Virtual Network Embedding* problem.

2.2 Why Do We Need Network Virtualization?

The major motivation of network virtualization is to address the impasse of the current Internet. To clearly understand that why network virtualization is introduced: *(i)* what is the cause of Internet ossification? *(ii)* Why can network virtualization address this impasse?

The fundamental causes of Internet stagnation lies on two facts. First, the stakeholders (i.e., ISPs such as Verizon, Sprint, owners of physical network infrastructure) have no economic incen-

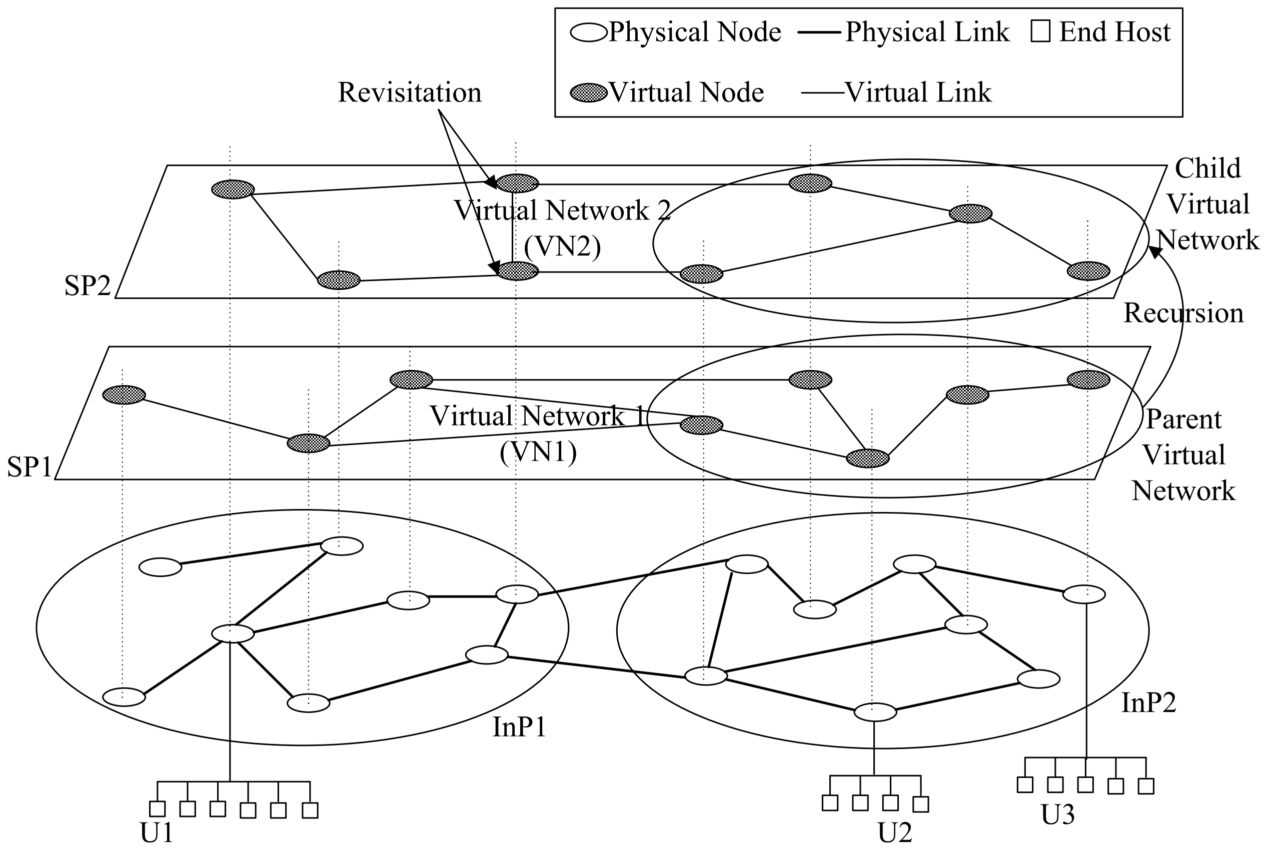


Figure 2.1. Network Virtualization Environment [1]

tive to adopt revolutionary technologies, given the potential risk and considerable CAPEX/OPE cost. Second, due to the design of the current Internet, employment of new technologies calls for agreement and coordination among multiple parties (e.g., technology innovators, hardware manufacturers, and ISPs), which, however, is difficult to achieve in reality.

The introduction of network virtualization can remove these barriers with the idea of creating a layer of abstraction through decoupling the traditional ISPs into the Infrastructure Providers (InPs), and the Service Providers (SPs) [2] [3] [4] [5]. After the decoupling, the SPs possess a virtualized view of the underlying network infrastructure and thus enjoy architecture-oblivious freedom of adopting revolutionary technologies [1] [4] [5]. Likewise, the InPs can transparently advance the physical network without service disruptions rippled to the SPs.

Finally, note that although network virtualization brings unprecedented flexibility for SPs and InPs to embrace revolutionary technologies, it is challenging to address the *virtual network embedding* problem.

2.3 Related Work

In this section, we review the recent advancements in network virtualization. As this dissertation is mainly related to three lines of research: virtual network embedding, survivable virtual network embedding, and optical network virtualization, we focus on the respective historic work.

2.3.1 Virtual Network Embedding

Virtual network embedding (VNE) process consists of two major components: node assignment, which is the mapping of the virtual node (with computational capacity requirement) to the substrate node; and link mapping, which is the mapping of the virtual link (with bandwidth capacity requirement) to the substrate path(s). VNE problem can be proven to be NP-Complete by simply showing that the node assignment problem alone is NP-Complete [8].

Given its NP-Completeness, VNE problem is normally reduced to integer linear programming (ILP) formulations. The optimal solution then can be obtained by solving the ILP formulations with ILP solvers (e.g., CPLEX [21], GLPK [22]). ILP-based solution, however, is intractable for large problem instances. For large scale instances or cases that timely VNE solution is needed, a alternative approach is to solve the relaxation of the optimal ILP model or seek heuristics or meta-heuristics. Therefore, we classify existing VNE approaches into four categories: optimal ILP solutions, ILP-based relaxations, heuristics, meta-heuristics, as shown in Table 2.1.

As shown in Table 2.1, a representative VNE ILP model is the link-based ILP model proposed in [9]. In this dissertation, we present a Path-based formulation for the VNE problem. On the one hand, path-based formulation generally possesses less number of variables than that of link-based ILP. On the other hand, as to be shown later, path-based ILP model lead to a simple and straightforward relaxation approach by limiting the path space to a selected set of paths.

Table 2.1. Four Types of VNE solutions

VNE Solution	Representative Work	Characteristics
Optimal ILP Solution	Link-based ILP [9]	intractable for large instances
ILP-based Relaxation	Rounding [9]	non-optimal, and no quality guarantee
Heuristics	[10] [23]	non-optimal, and no quality guarantee
Metaheuristics	[24] [25]	not optimal

Heuristic VNE solutions can be further classified as one-shot or two-step VNE approaches. In the former, the node assignment and link mapping are done in a coordinated manner or the same stage. For instance, as VNE process shares similar structure of sub-graph isomorphism, the authors of [10] present a backtracking approach based on a revised version of a subgraph isomerism scheme which handles node mapping and link mapping at the same stage. As the VNE process consists of two major components (node mapping and link mapping), a straightforward approach is to adopt a divide-and-conquer strategy that separates the node assignment and link mapping. In the first stage, all the node mapping are decided and fixed. In the second stage, based on the nodes in the prior stage, the link mapping is realized (e.g, as an instance of the multi-commodity network flow problem). A representative work that adopts this strategy is [23]. In this work, topology attributes are taken into account to rank the importance of each node. After calculating the the rank (importance) of the virtual and substrate nodes, each virtual node is sequentially mapped to the substrate node, both following the order of rank. After the node mapping is completed, link mapping is done by applying a shortest path algorithm or a multi-commodity flow algorithm due to whether path splitting is allowed. With a divide-and-conquer approach, it is inexorable, however, to sacrifice the solution quality since the node assignment decision is not holistic and not-reversible. We hence are motivated by the following question: Is it possible to maintain the *simplicity* of the *divide-and-conquer* strategy while still achieving *optimality* for the VNE problem? We answers this question with an intelligent decomposition framework supported by the *Primal-Dual* analysis of the proposed path-based ILP model.

Similar to regular heuristics, Metaheuristic solutions are not optimal as shown in Table 2.1. For Instance, [24] proposed an Ant-Colony-based algorithm to solve the VNE process where artificial ants are employed to explore the possible solution space. In [25], another population-based approach, namely particle swarm optimization, are proposed to address VNE via the evolution process of particles.

2.3.2 Survivable Virtual Network Embedding

With network virtualization gaining momentum, the concept of survivable network virtualization attracts considerable attention recently. Firstly, we classify the possible failures in network virtualization context into four categories as shown in Fig. 2.2. Note that a substrate node can be either a switching node which simply transits traffic flow, or a facility node which provide computing and/or storage capacities. Likewise, the substrate links can be access links (i.e, links between a switch and facility node) or transport links (i.e., links between switches nodes). By default, it is generally the focus to study the failure of the latter case. Figure 2.2 (a) shows an example of facility node failures. In this case, all the virtual nodes hosted by the failed substrate nodes are impacted. Figure 2.2 (c) contains an example for both access link and transport link failure, respectively. Figure 2.2 (b) shows the failure of a switch node. In this case, the attached facility node is logically removed from the network. Finally, Figure 2.2 (d) shows an example of reginal failure (e.g., due to earthquake) that affects a group of nodes and links.

The concept of survivable networking embedding was firstly introduced in [13], which focused on the protection of single link failure. Following that, various types of failure scenar-

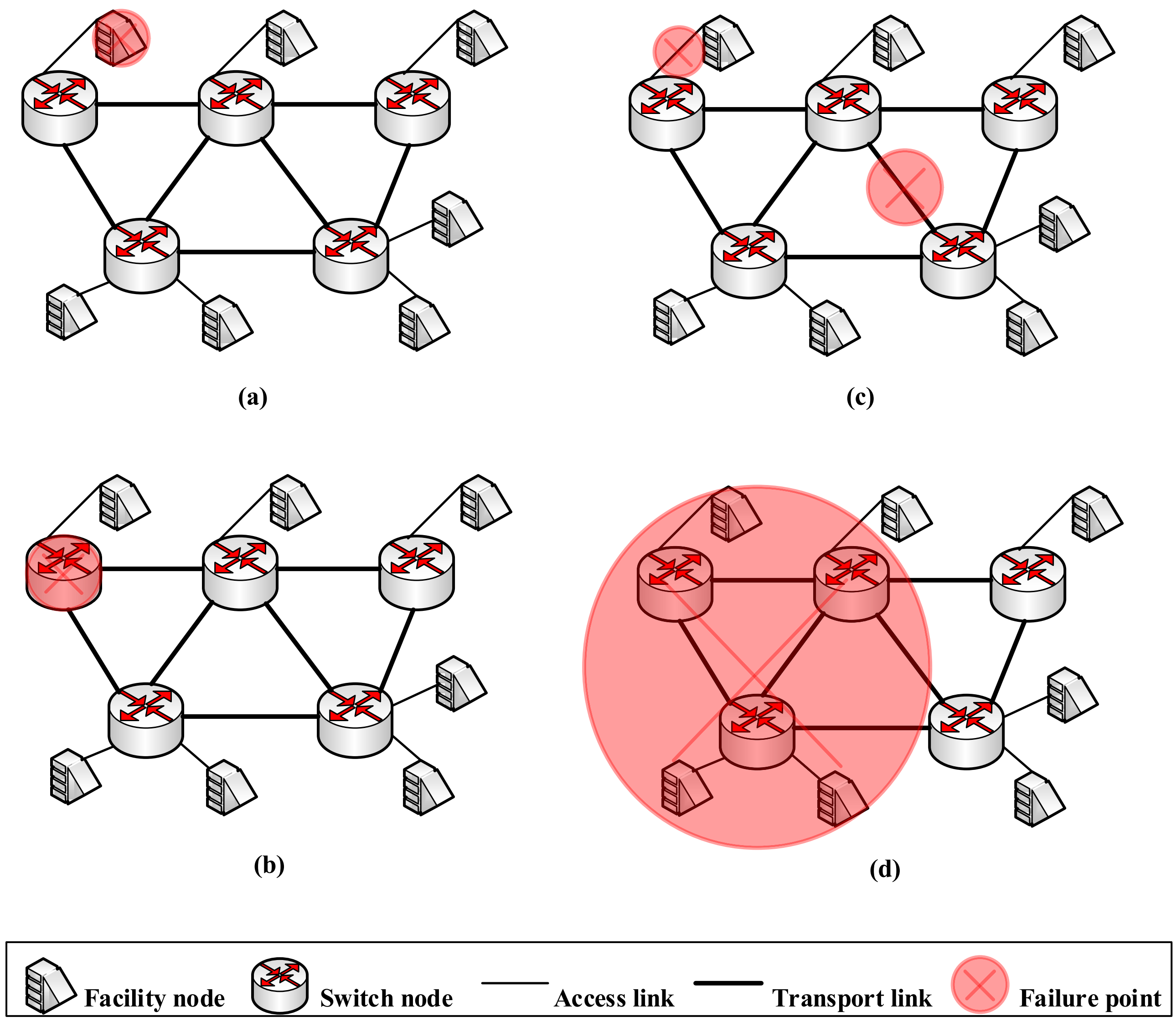


Figure 2.2. Different Failure Types

ios, including single link, single facility node and single regional failure have been investigated [15], [16], [17], [26]. The authors of [15] studied the protection of single regional failures in network virtualization. The basic idea of their approach is to re-map the virtual network for each regional failure assuming that the number of distinct regional failures are finite. This re-mapping process is done on the induced substrate networks where the links and nodes affected by the given regional failure are removed and hence result in a resilient mapping against the failure. In [16], the authors proposed a two-step method to enable the network survivability against the single facility node failure (i.e., the node with computing capability, which differs from the switch node that impacts the network connectivity). In the first step, an auxiliary graph of the virtual network is constructed, which can embed the network survivability by incorporating redundant nodes or links. For instance, to protect the failure of virtual node i, a backup node is added to the virtual network, say j, to replace i after the failure. Note that under the single failure assumption, node j can also be used to protect the failure of other virtual nodes for resource sharing. In the second step, the auxiliary graph is mapped to the substrate network (using existing mapping algorithms). The advantage of the two-step approach is the transparency of the protection process to the InPs. Another instance of the two-step method appeared in the study of [17]. Different from [16], the authors managed to further reduce the allocated backup resources with a *failure-dependent* strategy. Specifically, when Node i fails, the role of Node i may be replaced by any other nodes after a rearrangement of all the nodes (including the backup node(s)). This strategy is novel and interesting, however, it may not be practically applicable due to the large amount of possible migrations of working nodes/links. For a more detailed classification and discussion on various types of failures in network virtualization, one can refer to [27] and the references therein.

In general, we can classify the strategies of enabling survivability in virtualization into two categories. The first one is to provide protection at the physical network tier (i.e., for the switch nodes and links), which, fundamentally, has no difference with the protection schemes adopted in traditional networks. The second is based on the enhancement of the virtual network, where the facility node failure has to be carefully addressed. Among various node failure scenarios, in this dissertation, we target on enabling survivability under single facility node failure. Different from the existing work, in Part 5 we present a joint optimal model for the allocation of both active and backup resources, where both splittable and non-splittable mapping are taken into account.

2.3.3 Optical Virtualization

Optical networks are widely known for the high bandwidth and energy efficiency [18], [19]. The explosive growth of the Internet traffic clearly calls for the support of a bandwidth-abundant optical substrate [28]. To enable bandwidth-hungry application of the future Internet, optical network should be virtualized via abstraction, partition/aggregation of optical resources, and provide any-to-any bandwidth-abundant connectivity for the service layer of network virtualization in an elastic, agile and automated manner. At the same time, however, challenges including the heterogeneities of optical resources, and the physical constraints of optical transmission bring extra dimensions of complexity to the VNE problem.

In the literature, the idea of optical virtualization was initially introduced in [29]. It is shown in [29] that when automated optical path provisioning, segmentation/aggregation of network resources, and optical network resource management/coordination are all feasible, optical networks can be virtualized to provide stronger and smarter support to the upper layer. The authors of [30], from a practical viewpoint, discussed virtualization associated with major optical enabling technologies, devices, architectures. Specifically, the resource partition and aggregation for various optical resources including switch ports and link capacity are all investigated. With a virtualized optical network, various applications can benefit from the huge optical bandwidth, including data center and cloud computing applications, which is discussed in [31]. Recently, the authors of [32] studied the RWA problem in a virtualization context. Different from above work, in this dissertation, we introduce the concept of *Connectivity as a Service*, and investigate the *integration* of above two trends. Moreover, we present a holistic view of the motivations, architecture, and explicitly assess challenges on the way towards a virtualized optical substrate that supports network virtualization.

PART 3

THE OPTIMAL PATH-BASED FRAMEWORK FOR VNE

Given the NP-Completeness of the VNE problem, existing optimal solution for VNE is based on the link-based Integer Linear Programming (ILP). Related relaxation approaches are also developed based on the rounding of the ILP model. As discussed earlier, the first suffers from the extensive computational time of the ILP solver in practice, while the latter cannot provide a near-optimal solution with guarantee on closeness to the optimality. In this chapter, we fill these gaps by presenting a path-based Integer Linear Programming (ILP) model for virtual network embedding problem. We also discuss the importance of the location-awareness in network virtualizaition, which is incorporated in our model. Based on a compact path-based ILP model, our overall idea is to employ a *branch and bound* framework [33] to resolve the integrity constraints, while embedding the *column generation* process [34] to effectively obtain the lower bound for branch pruning.

The sections hereafter are organized as follows. We first introduce the importance of location information in network design in Section 3.1. We present the network model and the formal problem definition in Section 3.2. Then the path-based ILP model is presented in Section 3.3. Section 3.4 and Section 3.5 describe the column generation approach and the branch-and-bound framework respectively. The proposed framework is evaluated and analyzed in Section 3.6. Finally, we conclude this chapter in Section 3.7.

3.1 Location-awareness in Virtual Network Embedding

In network design, the location constraint is critical and can significantly affect the system performance. In the following, we discuss the impact of link location, node location information in network virtualization.

3.1.1 The Impact of the Link Location Information

With virtual network embedding, a virtual link is mapped to path(s) between the two substrate nodes where the respective two virtual ends reside. The link location information thus can be captured by the end-to-end propagation delay (i.e., latency) of the mapped path, which can be reflected in our model.

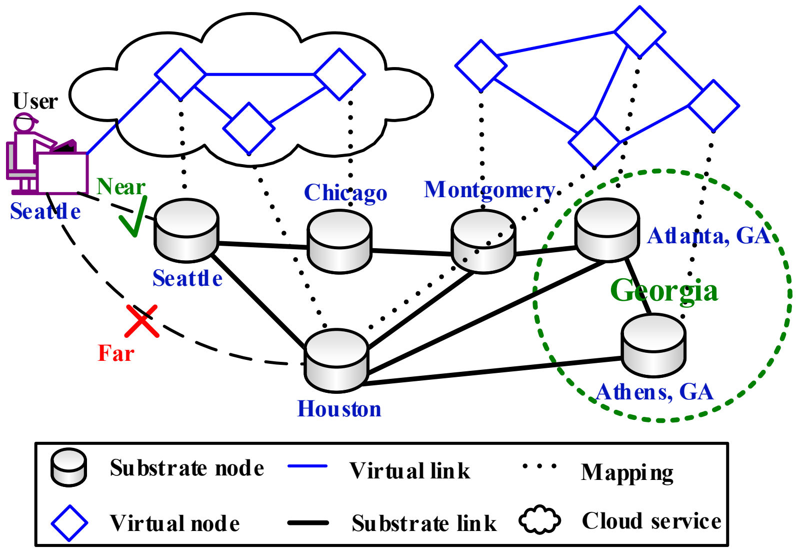


Figure 3.1. Possible Location Requirements in Network Virtualization

3.1.2 The Impact of Node Location Information

In virtual network embedding, a virtual node is hosted by a substrate node that has sufficient computing capacity. We illustrate a few scenarios where the location constraint plays a vital role in Fig. 3.1. In the first scenario, the service level agreement (SLA) may force the SP to place constraints on the location of the substrate nodes for the sake of network QoS metrics (e.g., response time). For instance, the Cloud user located at Seattle obtains a better user experience when the chosen computing/data center server resides in Seattle rather than Houston. Second, the location constraints may be posed due to the geographic requirements of the SP. As shown in Fig. 3.1, an SP headquartered in Atlanta may prefer their major computing servers located closely to the

Atlanta area, where their major customers reside. Other than above scenarios, the location information in resource backup is also critical and can impact the system performance to a large extent, as elaborated in [35]. In the model presented in the next section, we also take the node location constraint into account.

3.2 Network Model and Problem Definition

In this work, we model the virtual network as an undirected weighted graph $G^V = (N^V, L^V)$, where N^V is the set of virtual nodes with computing and location requirements, and L^V is the set of virtual links with bandwidth demands (and/or latency information). The computing resource requirement of Node a is denoted as $cr(a)$. The bandwidth demand of a virtual link (say $a - b$) reflects the bandwidth requirement between the two adjacent nodes (i.e., a and b) of the link, denoted by $br(a - b)$. For example, Figure 3.2(a) shows a virtual network consisting of three virtual nodes and three virtual links, and the numbers besides the nodes and links represent the computing and bandwidth requirements respectively. Figure 3.2(b) shows a virtual network with only two virtual nodes and one virtual link.

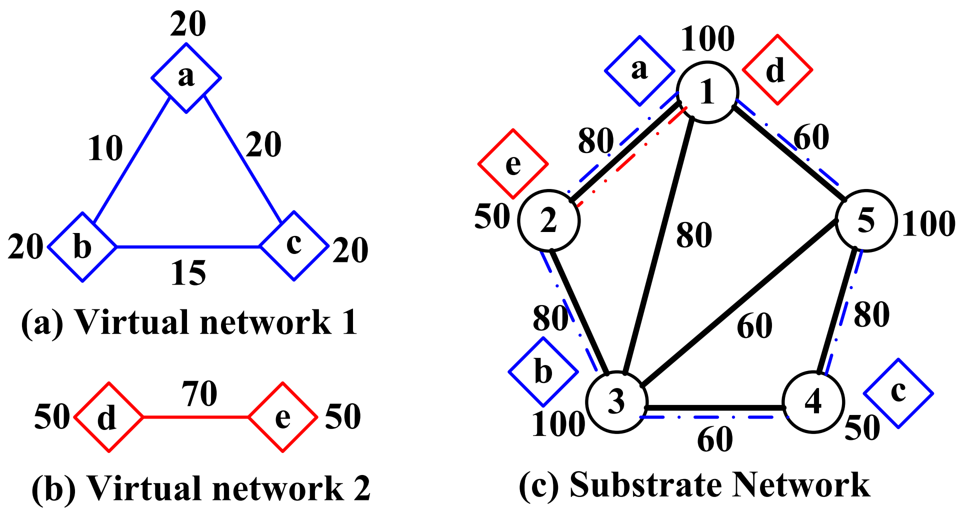


Figure 3.2. Embedding of Virtual Networks into a Substrate Network

Similarly, the substrate network is modeled as an undirected weighted graph $G^S = (N^S, L^S)$, where N^S is the set of substrate nodes, and L^S is the set of substrate links. In general, one virtual node is mapped onto one substrate node, and no two virtual nodes from the same virtual network can share the same substrate node [9]. In addition, one virtual link is mapped to path(s) between the two substrate nodes which hold the two virtual nodes of this virtual link. Figure 3.2(c) shows one possible mapping scenario of the two virtual networks onto a substrate network which consists of five substrate nodes, connected by seven substrate links. The numbers besides the nodes and links in Figure 3.2(c) represent the node and link capacities respectively.

Without loss of generality, we model the location constraint as a mapping $l : N^V \rightarrow N^S$. For a given virtual node, this mapping obtains all the substrate nodes that satisfy the location constraint. For instance, the location constraint of virtual node a may be denoted by a triple $lr(a) = (x_a, y_a, rad_a)$, where the coordinate (x_a, y_a) is the center (e.g., the headquarter of a company) of a circle, and rad_a is the maximum allowable distance from the center. Then the mapping l returns all the substrate nodes located within the above circle. We now formally define the *location-aware Virtual Network Embedding problem (VNE)* as an decision problem as follows.

Definition: **VNE Problem** - Given virtual network $G^V = (N^V, L^V)$, and substrate network $G^S = (N^S, L^S)$, can the virtual network be mapped to the substrate network while satisfying the follow requirements: *(i)* for each virtual node/link, it is mapped to the substrate network meeting the capacity/bandwidth constraint; *(ii)* for each virtual node/link, it is mapped to the substrate network meeting the location constraint.

To facilitate the ILP modeling in the next section, we view the virtual network embedding process as a *multi-commodity flow* problem (i.e., one commodity per virtual link). This is achieved with constructing the auxiliary graph (AUG) that couples the virtual and the substrate networks as shown in Fig. 5.2. In specific, for each virtual node (e.g., Node a in Fig. 3.2(a)), we connect it to a group of substrate network nodes (e.g., Node $1, 2$ for virtual node a in Fig. 5.2) which satisfies the node capacity constraint (i.e., $cr(a) \leq cc(1)$, and $cr(a) \leq cc(2)$), and the location constraint (i.e., Node $1 \in l(a)$, and Node $2 \in l(a)$). The set of the coupling links is denoted as AE. With this construction, a feasible flow between two virtual nodes of the auxiliary graph (e.g., Node a and b in Fig. 5.2) then maps the virtual link $a - b$ of Fig. 3.2(a).

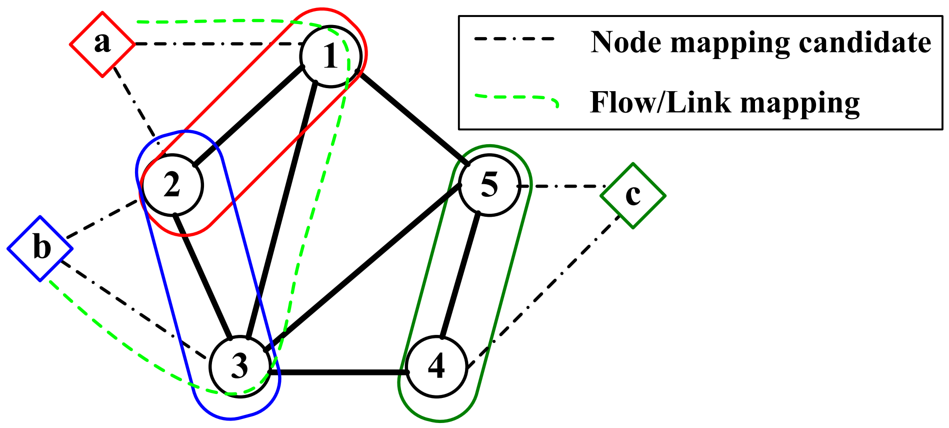


Figure 3.3. The Auxiliary Graph for the VNE Modeling

3.3 Path-based ILP Formulation for Virtual Network Embedding

In this section, we present the path-based Integer Linear Programming (ILP) model for the location-aware virtual network embedding problem. We refer to this model as the **P-VNE** model, with notations described in the table below.

f_p:	the flow on Path p, $f_p \geq 0$;
C_p:	the per unit flow cost of Path p;
$X_{I,i}$:	1 if virtual node I is mapped on to the physical node i, 0 otherwise;
μ_e:	the bandwidth capacity of physical link e;
P^k:	the set of paths for virtual link (or Commodity) k;
r_k:	the bandwidth requirement of virtual link (or Commodity) k;
$\delta_{p,(I,i)}$:	1 if link (I, i) is in Path p, 0 otherwise;
D_I:	the total bandwidth requirement of all incident virtual links of virtual node I;
c_e:	the per unit flow cost of Link e;
E_k:	The set of (two) end nodes of Commodity k;
P:	the set of all paths for all the commodities.

In this model, the objective is to minimize the overall cost of the flow, as shown in Eq. (6.1), where C_p is the per unit flow cost of path p, and $C_p = \sum_{e \in p} c_e$. Note that when we let $c_e = lat_e \forall e$ (where lat_e is the latency of the link e), the objective will be minimizing the overall latency. Alternatively, one can set $c_e = 1 \forall e$, then the objective is equivalent to the minimization of the overall allocated resources. In this work, the node resources are excluded in the objective since the consumed node resources are fixed under any mapping.

$$\min(\sum_{p \in P} C_p f_p) \tag{3.1}$$

The constraint shown in Eq. (3.2) ensures that the allocated resources upon each physical link is limited within the capacity bound.

$$\sum_{p: e \in p} f_p \leq \mu_e, \forall e \in L^S \tag{3.2}$$

To ensure that all the commodities (i.e., virtual links) are accommodated, the aggregated flow over all the paths of each commodity should be equal to the demand size of the commodity, which is reflected in Eq. (3.3).

$$\sum_{p\in P^k} f_p = r_k, \forall k \tag{3.3}$$

The constraints shown in Eq. (3.4) and Eq. (3.5) achieve the node mapping. In specific, Eq. (3.4) guarantees that no more than one virtual node resides in the same substrate node, while Eq. (3.5) ensures that each virtual node is mapped to exactly one substrate node.

$$\sum_{I:(I,i)\in AE} X_{I,i} \leq 1, \forall i \in N^S \tag{3.4}$$

$$\sum_{i:(I,i)\in AE} X_{I,i} = 1, \forall I \in N^V \tag{3.5}$$

Finally, a non-zero path flow passes through the auxiliary link (I, i) only when virtual node I is mapped onto physical node i, which is ensured by Eq. (3.6), where $D_I = \sum_{k:I\in E_k} r_k$. When $X_{I,i} = 0$, no path flow can exist on the auxiliary link (I, i).

$$\sum_{p\in P^k} \delta_{p,(I,i)} f_p \leq X_{I,i} D_I, \forall (I, i) \in AUG, I \in N^V, i \in N^S, (I, i) \in AE \tag{3.6}$$

It worth noting that: first, the size of the set P ($= \cup_k P^k$) can be exponential; second, not all the paths over the auxiliary graph can be included in the set P. For instance, for the commodity between node-pair (a, b) in Fig. 5.2, one cannot adopt the path going through other virtual nodes (e.g., the path a-1-5-c-4-3-b passing virtual node a). We name the path for a given node-pair that only going through substrate nodes (except the source and the sink node) as a *legitimate* path. In the formulation, we hence have to ensure all the paths in P are *legitimate* paths.

To deal with the first issue, we have two approaches in practice. In both ways, we firstly limit the number of chosen paths P' ($\subset P$). The resulted formulation is referred to as **P-VNE'**. One way is to directly solve the relaxed **P-VNE'** problem at the expense of the optimality. The other way, is to resort to the *column generation* approach and the *branch and bound* framework, as elaborated in the next sections. To tackle the second issue, we will show in the next section how to guarantee chosen paths are always *legitimate*.

3.4 Column Generation for The LP-VNE Formulation

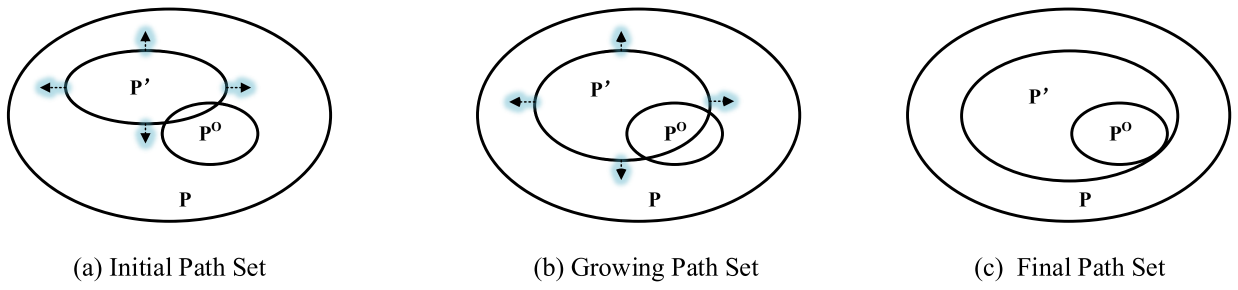


Figure 3.4. Path Set Growth Process

In this section, we present a *column generation*-based approach to solve the linear relaxation of the **P-VNE** problem, namely **LP-VNE**, which will serve as a building block for the *branch and bound* framework in the next section.

The idea of column generation is illustrated in Fig. 3.4. Although the set of all paths P can be exponential, the size of the path set that lead to the optimal solution to the **LP-VNE** problem (i.e., P^O) is limited. Hence, one can start with a small set of paths P' in Fig. 3.4(a), and incrementally incorporate new paths as in Fig. 3.4(b) until the optimal path set is contained as in Fig. 3.4(c) . In the process of path growth, there are two key decisions have to be made in each iteration: (i). Does the current path set P' contain P^O already (i.e., Fig. 3.4(c))? (ii) If not, which path should be included to further grow P' (i.e., Fig. 3.4(b))? We will resort to the *primal-dual* framework to address these two questions.

The dual of the **LP-VNE** problem, namely **D-LP-VNE** is presented below, including Eq. (4.5), Eq. (4.6), Eq. (3.9), Eq. (3.10). The variables y_e, λ_k, v_i, w_I and $\pi_{I,i}$ are the dual variables for the constraints of Eq. (3.2), Eq. (3.3), Eq. (3.4), Eq. (3.5) and Eq. (3.6), respectively. Particularly, the path reduced cost is reflected in Eq. (3.11) after a transformation of Eq. (4.6).

$$\max(\sum_k r_k\lambda_k - \sum_e \mu_e y_e + \sum_I w_I - \sum_i v_i) \tag{3.7}$$

$$-\sum_{e\in p} y_e + \lambda_k - \sum_{(I,i)\in p} \pi_{I,i} \leq C_p, \forall p \tag{3.8}$$

$$w_I - v_i + D_I\pi_{I,i} \leq 0, \forall(I,i) \in AE \tag{3.9}$$

$$y_e, v_i, \pi_{I,i} \geq 0, \lambda_k, w_I, \ \mathit{unrestricted} \tag{3.10}$$

$$\sum_{e\in p}(y_e + c_e) + \sum_{(I,i)\in p} \pi_{I,i} - \lambda_k \geq 0, \forall p \in P^k, \forall k \tag{3.11}$$

We note that a feasible solution $< f'_p >_{p\in P'}$ for **LP-VNE'** yields a feasible solution $< f_p >_{p\in P}$ for **LP-VNE** by setting $f_p = f'_p$ for $p \in P'$, and $f_p = 0$ for $p \in P \cap \overline{P'}$. Since $P' \subset P$, the value of the optimal solution of **OPT(LP-VNE')** is no less than that of **LP-VNE** (i.e., **OPT(LP-VNE')** ≥ **OPT(LP-VNE)**). In addition, due to the LP duality, we have the equivalent objective value when solving the primal and the dual problem optimally. Consequently, we have the relations among the objectives of above problems as summarized in Eq. (3.12), which will be used below to answer the first key question.

$$\textbf{OPT(D-LP-VNE')} = \textbf{OPT(LP-VNE')} \geq \textbf{OPT(LP-VNE)} = \textbf{OPT(D-LP-VNE)} \tag{3.12}$$

Before we present the exact column generation process, we firstly prove that the **LP-VNE** problem can be solved in polynomial time, as shown in Theorem 1 (although potentially there exists exponential number of paths). This fact can be established by showing that the dual of the **LP-VNE**, i.e., **D-LP-VNE** has a polynomial time separation oracle, which can take a given candidate solution and either confirm the feasibility or find a violated constraint in polynomial time.

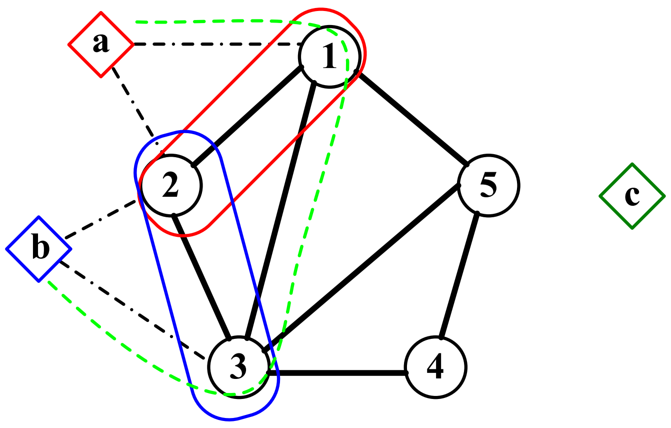


Figure 3.5. The Induced AUG for Node-pair (a, b)

This feature in turn indicates that the LP relaxation can be solved in polynomial time using the *Ellipsoid* algorithm [36] [37].

Theorem 1: *The **LP-VNE** can be solved in polynomial time.*

Proof: Equivalently, we show the existence of a polynomial time separation oracle. Clearly, Eq. (3.9) can be verified in polynomial time. We then only need to address Eq. (3.11). Given the dual variable π, y, and the cost c, one can construct an auxiliary graph with the edge cost $(y + c)$ for the substrate links, and π for the links between the virtual nodes and the substrate nodes. Then running the shortest path algorithm for each commodity (say k), and compare the weight of the shortest path with the value of the dual variable λ_k. If the weight $< \lambda_k$, the corresponding constraint is violated. If the shortest path of each commodity has the weight $\geq$ the respective value of dual variable λ, then the given dual solution is feasible. □

The column generation process addresses the first key question on new path selection by identifying paths (as the separation oracle) that cause the dual **D-LP-VNE** infeasible (i.e., with negative reduced cost). When no paths with negative reduced cost can be added to **D-LP-VNE'**, we have **OPT(D-LP-VNE')** = **OPT(D-LP-VNE)**. This conclusion guarantees that

OPT(LP-VNE') $=$ **OPT(LP-VNE)** based on Eq. (3.12), which indicates the optimality of the current path set P', and answers the first key question. In addition, to ensure that all the paths are *legitimate* paths, we construct the induced AUG for each commodity where all the other virtual nodes are isolated. For instance, when computing the shortest path for the node-pair (a, b), the virtual node c is disconnected in the corresponding subgraph shown in Fig. 3.5. In this way, one can guarantee the shortest path found on the subgraph is always legitimate for the respective commodity. The detailed column generation process is summarized in Algorithm 1.

Algorithm 1 Column Generation for LP-VNE

```
1:  repeat
2:      Solve the LP-VNE' with the current $P'$
3:      Obtain the dual variable $\pi$, $\lambda$, and $y$
4:      Assign $y + c$ as the edge weight of the substrate network
5:      Assign $\pi$ as the weight of the auxiliary edges between the virtual node and the substrate
        node
6:      for all Commodity $k$ of the virtual edge do
7:          Over the induced AUG of Commodity $k$, find a shortest $s_k$ -$d_k$ path, and let the weight of
            the selected path $p$ be $C_p$
8:          if $C_p < \lambda_k$ then
9:              $P' \leftarrow P' \cup p$
10:         end if
11:     end for
12: until No path has been added in the current iteration
```

Specifically, Algorithm 1 starts with solving the linear relaxation **LP-VNE'**[1], and obtain the dual variable in lines 2-3. The obtained values $y + c$, and π are used as the link weight for the auxiliary graph in lines 4-5. Now, to find a path with negative reduced cost, we can run the shortest path algorithm on the auxiliary graph between the source and the destination of the commodity (i.e., s_k, d_k), and compare the obtained shortest path cost (i.e., $\sum_{e \in p} (y_e + c_e)$) with the dual variable associated with the commodity of this path, i.e., λ_k. This process is repeated until no more path with negative reduced cost are found. In other words, no other path can increase the objective of the dual, and hence the optimality is established. Finally, the column generation process can be

[1]This can be done with ILP solver such as ILOG CPLEX [21].

embedded into the *branch and bound* framework [34] to prune the solution space and efficiently obtain the optimal solution for the VNE problem, which is further elaborated in the next section.

3.5 A Branch and Bound Framework for Optimal Virtual Network Embedding

Since the given **P-VNE'** (and **P-VNE**) contains binary variables, we rely on a *branch and bound* framework to address the integrity, and embed the *column generation* (CG) process to efficiently resolve the **LP-VNE'** problem for bounding. In the following, we first analyze the structure of the **P-VNE** formulation to facilitate branch pruning, and then present the detailed branch and bound framework.

3.5.1 Pruning based on the Structure of the P-VNE Formulation

The proposed ILP formulation contains boolean decision variable $X_{I,i}$. For each virtual node I, each substrate node i (with $X_{I,i} \in AE$) can lead to a branch for probing. Due to the structure of Eq. (3.4) and Eq. (3.5), one can significantly prune a large number of branches. For instance, in Fig. 3.6, when $X_{a,2} = 1$, then the decision variable $X_{a,1}$ can be pruned as $X_{a,1}$ must be 0 due to Eq. (3.5). We name this type of pruning as *Type I* pruning. In addition, when $X_{a,2} = 1$, the decision variable $X_{b,2}$ can be pruned in Fig 3.6 as it must be 0 due to Eq. (3.4). We name this type of pruning as *Type II* pruning. In general, once we chose the branch variable $X_{V,s} = 1$, we can prune all the branches for $X_{V,i}, i \neq s, (V, i) \in AE$ using Type I pruning, and $X_{I,s}, I \neq V, (I, s) \in AE$ with Type II pruning. Moreover, we resort to another way of pruning based the lower bound obtained from the column generation process, which is further discussed below.

3.5.2 The Branch and Bound Algorithm

The detailed framework is presented in Algorithm 2. In this branch and bound process, we maintain and update the best feasible solution and objective of the **P-VNE** (i.e., the upper bound), while branching on the integer variable $X_{I,i}$ for each virtual node I. Since the branch probing is done for each virtual node, the Type I pruning (Due to Eq. (3.5)) is implicitly applied (e.g., Line 4 of Algorithm 2). The Type II pruning enabled by Eq. (3.4) is achieved in Line 16. Specifically, if

Algorithm 2 *Branch* and *Bound* Algorithm for Virtual Network Embedding

```
1: Use a greedy algorithm to obtain an initial solution, namely $s_c$, assume that the associated object value is $v_c$
2: Solve the **LP-VNE** problem using CG, and let LB equal to the solution objective
3: Sort all the virtual nodes according to the node-degree in the auxiliary graph non-decreasingly, and name the resulted node set as $S_I$=$\{I_1, I_2, ..., I_n\}, n = |N^V|$
4: Push $\{(X_{I_1,j}, 1)\}$ to the stack for all the $j \in N^S$ and $(I_1, j) \in AE$
5: repeat
6:     Pop the stack, and update the active branch variable for each virtual node if exists
7:     if No more branch variable then
8:         Solve the resulted LP program using CG, assume that the resulted solution is $s_b$ with objective value $v_b$
9:         if $v_b < v_c$ then
10:            $v_c \leftarrow v_b$
11:            $s_c \leftarrow s_b$
12:        end if
13:        Continue
14:    end if
15:    Based on the order in $S_I$, decide the current branch variable, say $I_i$
16:    for each branch variable of $I_i$ (after Type II pruning decision), say $X_{I_i,j}$ ($j \in N^S$ and $(I_i, j) \in AE$) do
17:        Solve the **LP-VNE** using CG (after applying all the branch variable currently active: $I_1$-$I_i$), assume that the resulted solution is $s_b$ with objective value $v_b$
18:        if Case I: None feasible solution found then
19:            Continue
20:        else
21:            if Case II: $v_b > v_c$ then
22:                Continue
23:            end if
24:        else
25:            if Case III: $v_b < v_c$ and $s_b$ includes a binary assignment for Variable $X$ then
26:                $v_c \leftarrow v_b$
27:                $s_c \leftarrow s_b$
28:            end if
29:            if Case IV: $v_b < v_c$ and $s_b$ does not include a binary assignment for Variable $X$ then
30:                Push $(X_{I_i,j}, 1)$ to the stack
31:            end if
32:        end if
33:    end for
34: until Stopping Criteria is Reached
```

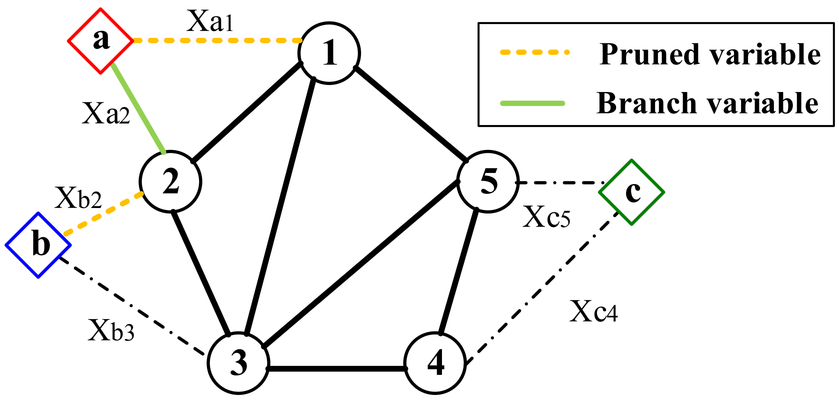


Figure 3.6. Example for Branch Variable and Pruned Variable

any virtual node with a higher order (than the current virtual node I_i) has been mapped to substrate node j (i.e., $\exists X_{A,j} = 1, A = 1, 2, ..., i-1$), the branch $X_{I_i,j}$ is disabled. Line 7 to 14 deals with the case that all the virtual nodes have a determined assignment, thus resulting a Linear Programming (LP) problem and no more probing is needed. Moreover, the column generation process is embedded to resolve the **LP-VNE** problem, which produces the lower bound for the corresponding **P-VNE** problem. The lower bound can help in pruning two types of branches: *(i)* that has no feasible solution (i.e., Case I in Line 18); *(ii)* that has the objective value greater than the solution found so-far (i.e., Case II in Line 21). We refer to above pruning as Type III pruning. For Case III in Line 25, we obtain a feasible solution to the **P-VNE**, thus the best solution and objective value can be directly updated. For Case IV where the **LP-VNE** problem returns a solution smaller than the current best solution, further probing is proceeded.

3.5.3 Stopping Criteria and Time Complexity

The above framework can achieve optimal virtual network embedding when the stopping criteria is *Empty of the Stack*. Theoretically, this criteria can lead to exponential time complexity. Due to the pruning discussed above, however, the computational time can be significantly reduced in practice. Given the need for a fast algorithm in cases such as on-line virtual network embedding, above framework can incorporate another stopping criteria as $(v_c - LB)/LB \leq \epsilon$ where v_c is the best solution so far (i.e., the upper bound), and ϵ is a tunable number. When ϵ is small, the probing process terminates when a solution with the expected quality is found. Note that the quality of the solution is guaranteed to be within $(1+\epsilon)$**OPT** for each problem instance. Another merit of this criteria is the flexibility in trade-offing computational time and solution quality via tuning the value of ϵ.

3.6 Performance Evaluation and Analysis

In this section, we evaluate the proposed scheme and compare the performance with other approaches. We are interested in three metrics: the computational time, the optimality of the results, and the request blocking probability in on-line VNE. We randomly generate the virtual network request with number of nodes ranging in $[2, 10]$. The size of the substrate network falls in the range of $[10, 50]$. Similar to [9], both the virtual and substrate network average connectivity is 50%. The bandwidth/computational requirement of the virtual network link/node is randomly generated within $[1, 20]$, while the link/node capacity of the substrate network is randomly generated within $[1, 50]$. The location constraint is also randomly generated to ensure that each virtual node has at least 2 substrate nodes that can be mapped to. Multiple thousands of the instances are simulated and the average performance is reported below.

3.6.1 Computational Time

Figure 3.7 presents the running time (in seconds) comparison among various approaches. The *Link-based ILP* is the formulation adapted from [9]. *P-VNE* refers to the optimal solution obtained with the approach proposed in this work. *P-VNE'(k=n)* refers to the path-based formulation with relaxed path variables, where k specifies the number of shortest paths for each commodity provided as the input. *P-VNE* ($\epsilon = n\%$) refers to the proposed branch and bound scheme with the parameter $\epsilon = n\%$.

From Fig. 3.7, one can see that among all the approaches, the *Link-based ILP* consumes the most computational time. The proposed *P-VNE* scheme can equivalently obtain the optimal result with reduced computational time. When limiting the value of k, one can significantly reduce the time. However, this is at the expense of the optimality. Particularly, when k is too small (e.g., $k = 1$), one may not obtain a feasible solution due to the limited path searching space. The approaches of *P-VNE* ($\epsilon = 10\%$), and *P-VNE* ($\epsilon = 5\%$), interestingly, can reduce the time complexity with guaranteed performance sacrifice (i.e., within $(1 + 10\%)$, and $(1 + 5\%)$ of the optimal solution, respectively), thus they can be more preferable in practice. In the next, we will look at the objective optimality comparison for above approaches to further understand the time-optimality tradeoff.

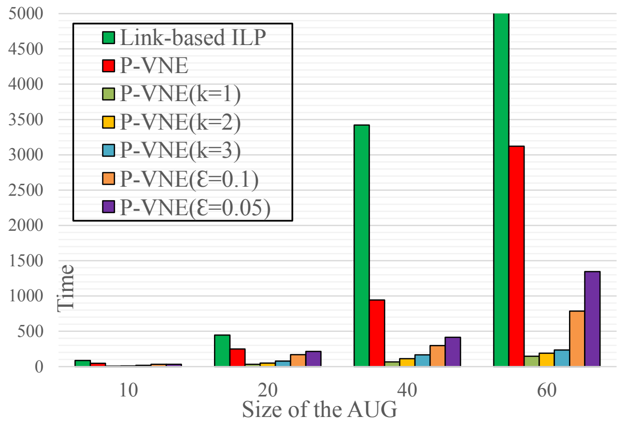


Figure 3.7. The Comparison of Computational Time

3.6.2 Objective Optimality

To compare the total consumed resources, we set $c_e = 1, \forall e$ assuming that all the resources have the same unit price. The comparison results are show in Fig. 4.3, where the X-axle is the size of the virtual network, and the Y-axle shows the objective value (i.e., total consumed resources) after solving the **VNE** problem using different approaches. In all the cases, we set the size of the substrate network to be 50. From Fig. 4.3, clearly, both the *Link-based ILP* and *P-VNE* approaches can obtain the optimal objective value in all the cases. The *P-VNE ($\epsilon = 10\%$)* and *P-VNE ($\epsilon = 5\%$)* can obtain near-optimal solution in all cases within $(1 + 10\%)$, $(1 + 5\%)$ of the optimal objective value, respectively. For the approach of *P-VNE'* with various k, one can see that increasing k

clearly improves the performance, but the increase from $k = 2$ to $k = 3$ is more evident than that of from $k = 1$ to $k = 2$.

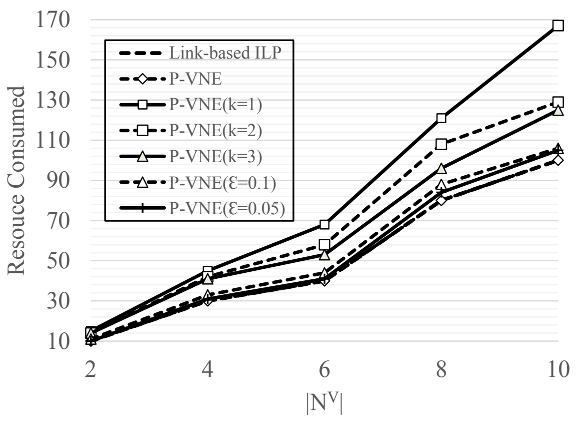


Figure 3.8. The Comparison of the Consumed Resources

3.6.3 Request Blocking Probability

In on-line virtual network embedding, virtual network can arrive and depart dynamically. It is thus interesting to see the blocking probability when employing above approaches. To simulate a dynamic traffic scenario, we assume that virtual network requests arrive as a *Poisson* process with an average rate of 4 requests per 100 time units with exponentially distributed duration of 1000 time units on average. The comparison is show in Fig. 3.9, where the X-axle is the time unit of the simulation, and the Y-axle shows the averaged blocking probability using different approaches. Clearly, the approaches of *P-VNE'* with $k = 1, 2, 3$, although consume less computational time (as shown above), have more request blocking ratio compared to the rest. This is mainly due to the limited path space when searching for resources to accommodate dynamic requests. The *Link-based ILP* and *ILP P-VNE* outperform relaxation approaches with *P-VNE* ($\epsilon = 10\%$), *P-VNE* ($\epsilon = 5\%$). Due to the dynamic nature of the traffic demand, however, this advantage may not be obvious or always existing(e.g., at Time Unit 20000). Note that for on-line scheduling, the computation time may have higher priority than the optimality. As a result, the approaches of *P-VNE* ($\epsilon = n\%$) or even the *P-VNE'* with k may be adopted in practice for timely decision or when resources are abundant.

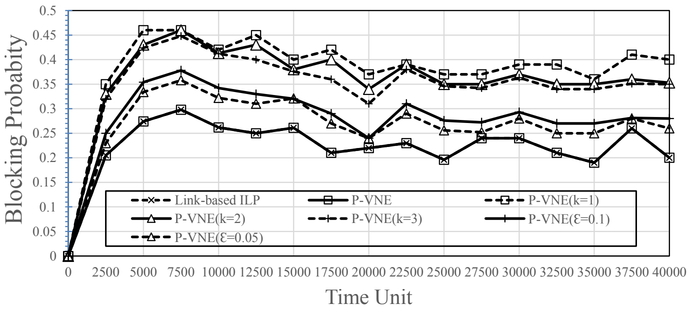


Figure 3.9. The Comparison of Request Blocking Probability

3.7 Remarks

In this chapter, we have presented a path-based ILP model for the VNE problem. A column generation approach incorporated into a branch-and-bound framework has been designed to achieve an optimal solution or a near-optimal solution with quality guarantee (i.e., within $1 + \epsilon$ factor of the optimal solution). Our performance evaluation has shown the correctness and effectiveness of our ILP model and the overall framework.

PART 4

VNE: A DECOMPOSITION APPROACH

The VNE process generally consists of two components: the node assignment, which is the mapping of the virtual node (with computational capacity requirement) to the substrate node; and link mapping, which is the mapping of the virtual link (with bandwidth capacity requirement) to the substrate path(s). In a one-shot mapping scheme such as the path-based formulation presented in the prior chapter, these two sub-problems are resolved in a coordinated manner.

Given the nature of the VNE problem, an alternative strategy in VNE algorithm design for relaxation and heuristics is to decompose the problem into two sequential sub-problems [38] [39] [9] [40] [23] [41]: node assignment (NA), and link mapping (LM). The main idea is illustrated in Fig. 4.1. Specifically, in Phase I, the node assignment (NA) is decided using a greedy policy (e.g., *node ranking* algorithm in [23]) or the rounding solution of the VNE LP-Relaxation (e.g., [9]). In Phase II, when node assignment is fixed, the link mapping problem is reduced to a classic *multi-commodity flow* problem [42]. With this approach, it is inexorable to sacrifice the solution quality since the node assignment decision is not holistic and not-reversible.

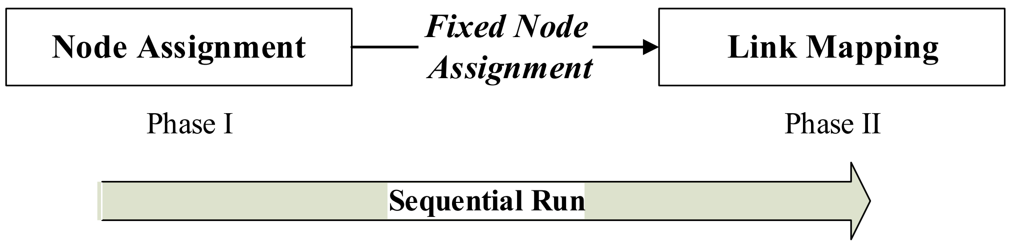


Figure 4.1. Sequential Node Assignment and Link Mapping

In our study, we are motivated by the question: Is it possible to enjoy the simplicity of a *Divide-and-Conquer* or *Decomposition* approach while maintaining the optimality (or guaranteed near-optimality) for the VNE problem? Fortunately, we confirm that the answer is positive when the NA and LM sub-problems can be *intelligently* integrated. Our major idea is illustrated in Fig. 4.2. Specifically, we construct the NA and LM sub-problems based on the *Primal-Dual* analysis of the path-based ILP model. The resulted two subproblems can feedback each other by producing lower bound and upper bound to the original problem, respectively. Eventually, when the lower bound and upper meet (or approach) each other, an optimal (or near-optimal) solution can be found for the original VNE problem.

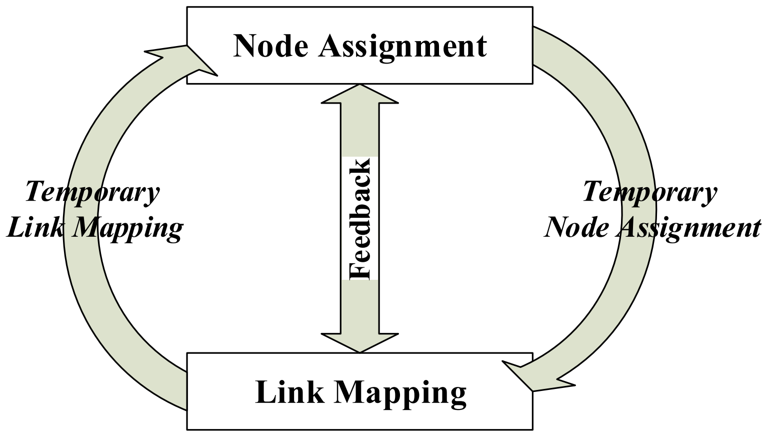


Figure 4.2. Integration of Node Assignment and Link Mapping

The remaining part of this chapter is organized as follows. Section 4.1 presents the link mapping sub-problem. In Section 4.2, we present the node assignment sub-problem. Section 4.3 presents the framework that incorporates the link mapping and node assignment sub-problem. Section 4.4 reports the performance study and analysis. Finally, we conclude this chapter in Section 4.5.

4.1 The Link Mapping Sub-problem

In this section, based on above the Path-based ILP model presented in Chapter 3, we present the link mapping sub-problem.

The link mapping sub-problem can be obtained assuming that the node assignment is already known. In the scope of our Path-based VNE model, we can simply fix all binary variable vector $X_{I,i}$

at a given value, say $\overline{X_{I,i}}$ [1], which results in a linear programming (LP) sub-problem (LM-Primal), as shown below. Note that since $\overline{X_{I,i}}$ is considered as a constant vector after fixing the value, the resulting problem does not have the constraints corresponding to Eq. (3.4) and Eq. (3.5). The resulting LM-Primal model still contains exponential number of path variables, which, however, can be optimally resolved in polynomial time using a *Column Generation* approach [43]. Also note that since we fixed the value of $X_{I,i}$, the solution of the LM sub-problem serves as an *upper bound* for the original problem.

$$\min(\sum_{p\in P} C_p * f_p) \tag{4.1}$$

s.t.

$$\sum_{p:e\in p} f_p \le \mu_e, \forall e \in L^S \tag{4.2}$$

$$\sum_{p\in P^k} f_p = r_k, \forall k \tag{4.3}$$

$$\sum_{p\in P} \delta_{p,(I,i)} * f_p \le \overline{X_{I,i}} * D_I, \forall (I,i) \in AE, I \in N^V, i \in N^S \tag{4.4}$$

4.2 Node Assignment (NA) Sub-problem

In this section, we further present the link mapping sub-problem based on primal-dual analysis of the Path-based ILP model presented in Chapter 3. Note that to ensure the *feedbacks* between LM and NA sub-problems, it is more complex to obtain the NA sub-problem.

We firstly obtain the dual formulation (namely LM-Dual) of the Link Mapping sub-problem, as shown in Eq. (4.5), and Eq. (4.6). In specific, function $f(\theta, \overline{X_{I,i}})$ is used to represent the objective of the LM-Dual formulation, where vector θ represents the dual variable y_e, λ_k, $\Pi_{I,i}$, of Eq. (3.2), Eq. (3.3), and Eq. (3.6) respectively, and $\overline{X_{I,i}}$ is the current value of node assignment vector $X_{I,i}$.

[1]However, this vector value should ensure a valid node assignment as specified in Eq. (3.4) and Eq. (3.5).

$$\max f(\theta, \overline{X_{I,i}}) = \sum_{k} r_k * \lambda_k - \sum_{e} \mu_e * y_e - \sum_{I,i} \Pi_{I,i} * \overline{X_{I,i}} * D_I \tag{4.5}$$

s.t.

$$-\sum_{e \in p} y_e + \lambda_k - \sum_{(I,i) \in p} \Pi_{I,i} \leq C_p, \forall p \tag{4.6}$$

One important feature of the LM-Dual LP formulation is that the *constraints* of this problem are independent of the fixed valued of $\overline{X_{I,i}}$. In other words, when changing $X_{I,i}$ at different values, the LP problem resides in the same LP feasible region bounded by Eq. (4.6). Different value of $X_{I,i}$, hence only leads to different extreme-point solution of the feasible region [44]. Let the set of all extreme points to be Ex, and $|Ex| = J$. Consequently, the LM-Dual formulation is equivalent to the following LM-Dual-EP formulation:

$$\max_{x \in Ex} f(x, \overline{X_{I,i}}) \tag{4.7}$$

Due to the strong duality [44], we know that LM-Dual-EP is also equivalent to the LM-Primal formulation, which relaxed the node assignment variable $X_{I,i}$ in the original VNE problem. If we reversely allow $X_{I,i}$ to take any valid value in LM-Dual-EP, the resulting formulation in Eq. (4.8) hence should lead to an optimal solution to the original VNE problem:

$$\min_{X_{I,i}:\ valid\ NA} \max_{x \in Ex} f(x, X_{I,i}) \tag{4.8}$$

Since it is a *min-max* problem, we further transform the Eq. (4.8) into the following for-m, namely NA-Min-Max, where Eq. (4.11), and (4.12) ensure that the solution is a valid node assignment.

$$\min z \tag{4.9}$$

s.t.

$$z \geq f(x_j, X_{I,i}), \forall x_j \in Ex, j = 1, 2, ..., J \tag{4.10}$$

$$\sum_{I \in N^V} X_{I,i} \leq 1, \forall i \in N^S \tag{4.11}$$

$$\sum_{i \in N^S, (I,i) \in AE} X_{I,i} = 1, \forall I \in N^V \tag{4.12}$$

In reality, it is impossible or time-prohibitive to obtain all the extreme points (i.e., $J = |Ex|$) for Eq. (4.10). However, every time when a LM sub-problem (based on a given $\overline{X_{I,i}}$) is solved, one can append a new extreme point (found in the LM-Dual formulation) to the NA-Min-Max formulation. Note that the resulting NA-Min-Max formulation with $J < |Ex|$ can be considered as the *Node Assignment* sub-problem since which relaxes the number of constraints, and produces the solution for the node assignment (i.e., $X_{I,i}$). Also, since $J < |Ex|$, the obtained objective is a *lower bound* of the original VNE problem.

4.3 A Decomposition Framework for Virtual Network Embedding

Relying on the obtained LM and NA sub-problem, we present in this section a framework that can generate optimal or near-optimal solution to the VNE process. The idea of this framework is to iterate over above two sub-problems: the LM sub-problem assumes fixed node assignment, and generates an upper bound to the original problem in each round. In addition, in each round, it leads to a new extreme point in solving LM-Dual formulation, and in turn a new constraint for the NA-Min-Max formulation for the NA sub-problem. Likewise, in each round, the NA sub-problem generates a lower bound for the original VNE problem, and produces a new solution of node assignment to feedback to the LM sub-problem. The optimality is reached when the lower bound meets the upper bound. Above process, is very similar to the *Bender's Decomposition* approach [45], [46] where the original problem is decomposed into two sub-problems, namely master problem, and sub-problem.

The overall process is shown in Algorithm 1. Basically, in each step, we solve the NA sub-problem to obtain a new value for the fixed variable vector $X_{I,i}$, and then solve the LM at the new value, which further leads to a new constraint to the NA sub-problem. When LB and UB generated

Algorithm 3 Iterative Decomposition Approach for VNE

```
1: Choose the first $\overline{X_{I,i}}$ for the original problem
2: Solve the LM sub-problem at $\overline{X_{I,i}}$
3: Set the $UB$ to the objective value of the LM sub-problem
4: $LB = -\infty$
5: $j = 0$
6: while $UB > LB$ ($or \frac{UB-LB}{LB} > \epsilon$) do
7:     $j = j + 1$
8:     Add $z \geq f(x_j, X_{I,i})$ as a constraint to the NA problem, where $x_j$ is the new extreme point
9:     Solve the new NA problem, and use its solution as the new $X_{I,i}$, and the objective value as
       the new LB
10:    Solve the LM sub-problem at $\overline{X_{I,i}}$, use its objective value as the new UB (if < current UB),
       and obtain a new extreme point by solving the LM-Dual
11: end while
```

in the NA and LM sub-problems meet[2], the algorithm terminates with the optimal solution. Moreover, we incorporate another stopping criteria to terminate the iteration when sufficiently good solution is found (i.e., $\frac{UB-LB}{LB} \leq \epsilon$). This can reduce the computational time, while ensures that the quality of the solution is within $(1+\epsilon)$**OPT** for each problem instance. Alternatively, one can also limit the number of iterations with $j \leq Iter\_No$, where $Iter\_No$ is the maximum number of iterations allowed. Note that there are multiple merits in our design: *(i)* With fixing the node assignment, the resulting LM sub-problem has a Linear Programming model, which can be optimally resolved in polynomial time; *(ii)* Different from approaches such as Lagrange relaxation [42], our approach always maintains feasible solutions to the original problem, and can lead to optimal solution if time allows; *(iii)* Bearing both the upper and lower bounds, our approach supports the flexibility for early termination with guaranteed solution quality.

4.4 Performance Evaluation and Analysis

In this section, we evaluate the proposed scheme and compare the performance with other approaches. We randomly generate the virtual network request with number of nodes ranging in $[2, 10]$. The size of the substrate network falls in the range of $[10, 50]$. Similar to [9], both the

[2]which is guaranteed to happen since the LB is increased in each iteration [45] [46].

virtual and substrate network average connectivity is 50%. The bandwidth/computational requirement of the virtual network link/node is randomly generated within $[1, 10]$, while the link/node capacity of the substrate network is randomly generated within $[1, 20]$. For comparison, we adapt the Link-based VNE formulation from [9] and refer to it as *Link-ILP-OPT*. The proposed Decomposition approach can lead to the optimal solution for the original problem, when the sub-problem is optimally solved (using column generation [43]). We refer to this approach as *D-OPT*. When the stopping criteria $\frac{UB-LB}{LB} \leq \epsilon$ is adopted, the resulting approach is called *D-Relax-ϵ*. In addition, we can directly solve the proposed Path-based ILP model selecting k-shortest paths for each commodity, and the resulting approach is named *Path-ILP(k=n)* when n shortest path are used for each commodity.

4.4.1 The Total Resource Consumption

To compare the total consumed resources, we let $c_e = 1, \forall e$ in the objective function, and set the size of the substrate network to be 50. As shown in Fig. 4.3, the X-axis is the size of the virtual network, while the Y-axis shows the total consumed resources after solving the **VNE** problem using different approaches. Clearly, both the Link-based ILP and proposed decomposition approach (without relaxation) can obtain the optimal value in all the cases. The relaxed decomposition approach with $\epsilon = 10\%$ leads to close to optimal solution in all cases within $(1 + \epsilon)$ of the optimal value. For the decomposition approach with various k, one can see that increasing k clearly improves the performance due to the increased path space for the traffic accommodation.

4.4.2 QoS Performance

One can also rely on the proposed approach to obtain a VNE solution with the best QoS metrics. To evaluate the QoS performance, we set the $c_e = lat_e$, i.e., the latency of the link e, and randomly generate the link latency within $[0.1, 1](ms)$, and the obtained performance for all approaches is shown in Fig. 5.11. Unsurprisingly, the least latency can be observed for the optimal solution from the Link-based ILP model as well as the un-relaxed decomposition approach. Again, decomposition approach with $\epsilon = 10\%$ leads to near optimal solution with a guaranteed closeness

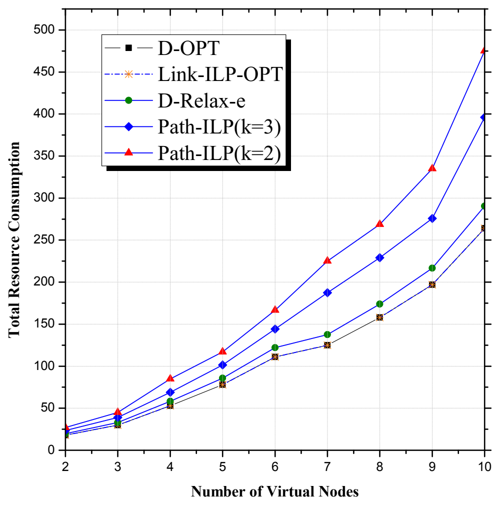


Figure 4.3. Total Consumed Resources

to the optimal solution.

4.5 Remarks

The role decoupling of the infrastructure provider (InP) and the service provider (SP) enables a virtualized view of the underlying network infrastructure to the SPs as well as the freedom of adopting the technological advancement to both. At the same time, however, network virtualization brings the overhead of solving virtual network embedding (VNE) problem. Existing VNE approaches are either time-prohibitive or non-optimal (particulary when Divide-and-Conquer or decomposition strategy is employed in the design). In this chapter, based on a Path-based Integer Linear Programming model, we design a novel decomposition framework to address the VNE problem. Based on the *Primal-Dual* analysis of the path-based ILP model, the proposed iterative process allows feedback between the Link Mapping sub-problem and the Node Assignment sub-problem, thus leading to an optimal solution to the VNE problem or achieve a near-optimal solution with *per-instance guarantee* on the closeness to the optimality in a timely manner. For the future work, we plan to extensively evaluate the proposed approach on large instances, and extend our idea to a survivable virtualization context.

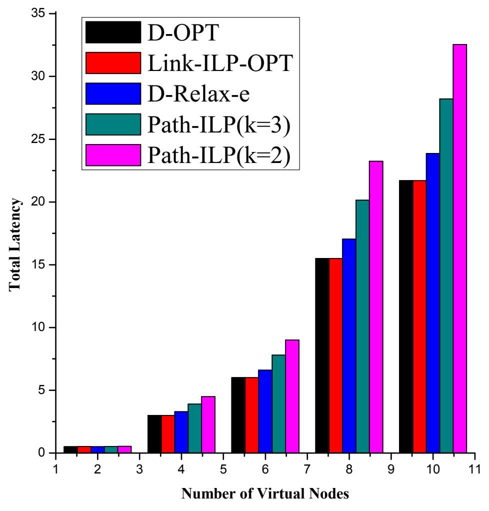


Figure 4.4. Latency Comparison

PART 5

SURVIVABLE NETWORK VIRTUALIZATION

With network virtualization gaining momentum, the concept of survivable network virtualization attracts considerable attention recently. The problem of *survivable virtual network embedding* (SNE) generally has to deal with failures of variety of network elements, including single link, single facility node and single regional failure. In this chapter, we study survivable network embedding with single facility node failure from a network flow viewpoint. The resulted problem, namely *survivable network embedding for single facility node failure (SNF)* has the added complexity of accommodating both the active and backup requests. Different from prior work, we take both splittable and non-splittable flow mapping into account, and present a joint optimal solution to both the working and backup resource allocation.

The remainder of this part is organized as follows. Section 5.1 presents the network model and formal problem definition. In Section 5.2, we elaborate the network flow view of the SNF problem, which leads to the MILP model in Section 5.3. We design Tabu-search based heuristic algorithms for the SNF problem under splittable and non-splittable flow scenarios in Section 5.4. Performance evaluation and analysis are given in Section 5.5. Finally, we conclude this chapter in Section 5.6.

5.1 Network Model and Problem Definition

In this section, we present the network model, and the definition of the survivable network embedding problem.

5.1.1 Network Model

Similar to earlier chapters, the virtual network request is denoted as an undirected weighted graph $G^V = (N^V, L^V)$, where N^V is the set of virtual nodes, and L^V is the set of virtual links. The

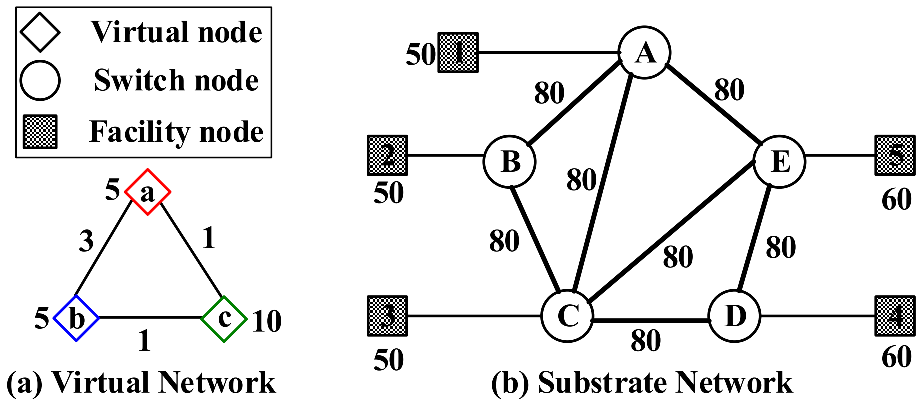


Figure 5.1. Virtual Network and Substrate Network

computing resource requirement of a virtual node a is denoted as $cr(a)$, while the communication demand of a virtual link (between virtual node a and b) reflects the bandwidth requirement between the two incident nodes (i.e., a and b) of the link, denoted by $br(a, b)$.

The substrate network is represented by an undirected weighted graph $G^S = (N^S, L^S)$, where L^S is the set of substrate links, and N^S consists of two types of nodes: the *facility* node and the *switch* node. The former is provisioned with the computing capability and hosts the virtual node, and the latter connects facility nodes to form the substrate network. We denote the available computing capacity of facility node I as $cc(I)$, and the available bandwidth of a substrate link a-b as $bc(a, b)$. An example of virtual network and substrate network is shown in Fig. 6.5(a), and 6.5(b), respectively.

5.1.2 Splittable Mapping vs. Non-splittable Mapping

In network embedding, the virtual node is mapped to the substrate facility node with sufficient computing capability. The virtual link, as a result, is mapped as the path(s) of the substrate networks. The non-splittable link mapping restricts that one virtual link is mapped to exactly one substrate path (with sufficient bandwidth on each hop). In contrast, in the splittable mapping, one virtual link can be mapped to one or multiple substrate paths (with sufficient aggregated bandwidth). In this work, we consider both mapping strategies and comparatively study their performance.

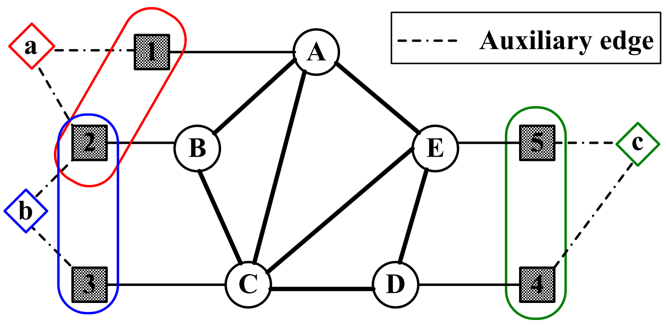


Figure 5.2. Auxiliary Graph

5.1.3 Problem Definition

We focus on the protection of the single facility node failure in this dissertation. Note that the failure of the switch node is not included in this dissertation since which can be addressed with protection schemes adopted in traditional optical networks. We formally define the Survivable Network embedding for single Facility node failure (SNF) problem as follows.

Definition 2: **SNF Problem.** Given the virtual network $G^V = (N^V, L^V)$, and substrate network $G^S = (N^S, L^S)$, the SNF problem is a decision problem that determines whether it can map the virtual network to the substrate network satisfying the following constraints: *(i)* for each virtual node/link, it is mapped to the substrate network meeting the capacity/bandwidth constraint; *(ii)* the virtual network is protected against any single facility node failure of the mapped substrate node.

Furthermore, we refer to the SNF problem with splittable flow as **SNS** problem, and the SNF problem with non-splittable flow as **SNN** problem hereafter.

5.2 The Network Flow View of SNF Problem

In this section, we present our network flow view of the SNF problem, and elaborate the resource sharing in the SNF problem.

5.2.1 Auxiliary Graph and Node Mapping

Similar to earlier chapters, we view the survivable network virtualization process as a *multi-commodity flow* problem via constructing an auxiliary graph (AUG) that couples the virtual and the substrate networks as shown in Fig. 5.2. In specific, for each virtual node (e.g., Node *a* in Fig. 6.5(a)), using auxiliary edges, we connect it to a group of substrate network nodes (e.g., Node $1, 2$ for virtual node *a* in Fig. 5.2) which satisfies the node capacity constraint (i.e., $cr(a) \leq cc(1)$, and $cr(a) \leq cc(2)$). In this way, the node capacity is automatically satisfied. For the virtual link, it then can be mapped as the feasible flow(s) between two virtual nodes of the auxiliary graph. Clearly, non-zero flow on an auxiliary edge indicates the mapping of the incident virtual node onto the incident facility node. We denote the set of auxiliary edges as L^A, and assign unbounded capacity to those auxiliary edges.

5.2.2 Link Mapping for the SNF Problem

For the protection of the facility node failure, we need to allocate both a primary and a backup facility node for each virtual node. Consequently, for a given virtual link, say *a*-*b*, one will observe **three** flows associated with it: the flow between the primary node of *a* and *b*; the flow between the primary node of *a* and the backup node of *b*; and the flow between the backup node of *a* and the primary node of *b*. The latter two flows are provisioned to cope with the failure of *a*, and *b*, respectively.

Further note that the splittable mapping and non-splittable mapping are reduced to the splittable and non-splittable flow mapping in the network flow model, respectively. In the former, the network flow can be carried by more than one paths over the AUG, while the latter can only send the flow over a single path of the AUG.

5.2.3 Resource Sharing

The network flow model also has to take two types of resource sharing into consideration: the sharing over the link bandwidth resources, and the sharing among the backup node resources. We discuss both types below and present detailed formulations in the next section.

The link bandwidth sharing can further be classified into two categories. Figure 5.3 shows an example for both of them. The virtual network to be mapped is the same with Fig. 6.5(a). The working paths(flows) and backup paths(flows) between nodes in Fig. 5.3 are denoted by

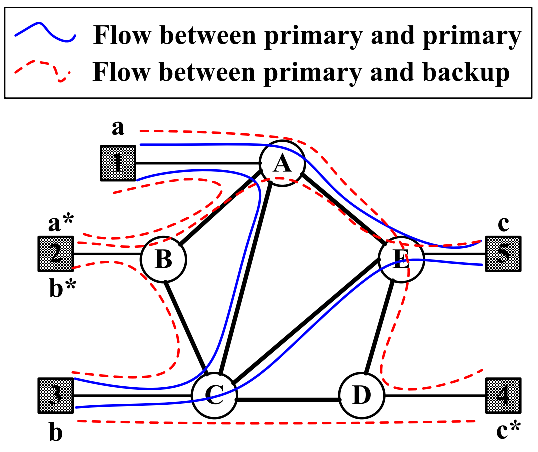

Flow between primary and primary
Flow between primary and backup
a
a*
b*
b
c
c*
1
2
3
4
5
A
B
C
D
E

Figure 5.3. Sharing in the SNF

solid curves and dashed curves respectively. After embedding, the virtual node a, b, and c are accommodated on Node 1, 3, and 5 respectively with their backup nodes $a*$, $b*$, and $c*$ residing on Node 2, 2 and 4. The path for virtual node pair (a, c) is 1-A-E-5, while the corresponding backup path for virtual node pair $(a, c*)$ is 1-A-E-D-4. These two paths both run through the edge (A, E). Meanwhile, the path for virtual node pair $(a*, c)$ (i.e., 2-B-A-E-5) also go through the edge (A, E). Sharing that happens at Edge (A, E) for the above three paths are in two categories: the sharing between paths for (a, c) and $(a*, c)$/$(a, c*)$ is the *working-backup* sharing; the sharing between paths for $(a*, c)$ and $(a, c*)$ is the *backup-backup* sharing.

The second type of sharing, node resource sharing can happen between the backup nodes of different virtual nodes. Above example shows the sharing of backup nodes between virtual node a and b since both nodes are protected by substrate node 2. Consequently, the backup computing resources of a and b are shared with each other.

5.3 ILP Model for SNS Problem and SNN Problem

We next present the ILP model for the optimization version of the SNS and SNN problems, which employ the following variables.

$w_{u,v}^{I,J}$:	the amount of working flow on the edge (u, v) (of the auxiliary graph) for the virtual edge (I, J);
$b_{u,v}^{I,J}$:	the amount of backup flow on the edge (u, v) (of the auxiliary graph) for the virtual edge (I, J) after the failure of virtual node I;
$W_{U,v}$:	for the auxiliary edge, 1 if virtual node U's primary node is mapped on to the physical node v, 0 otherwise;
$B_{U,v}$:	for the auxiliary edge, 1 if virtual node U's backup node is mapped on to the physical node v, 0 otherwise;
$r_{u,v}$:	the consumed backup resource on a physical link (u, v);
γ_v:	the consumed backup resource on a facility node v.
$x_{u,v}^{I,J}$:	1 if physical edge (u, v) is carrying the working flow of virtual edge (I, J), 0 otherwise;
$y_{u,v}^{I,J}$:	1 if physical edge (u, v) is carrying the backup flow of virtual edge (I, J), 0 otherwise;

Since the virtual network is undirected, the active flow from I to J is the same flow from J to I. In the formulation, we adopt the convention that the variable for the flow between virtual node I, J (i.e., $w_{u,v}^{I,J}$) can be nonzero only when the node ID of I is less than J. However, this is not the case for backup flow. In specific, the backup flow variable $b_{u,v}^{I,J}$ indicates the flow between the virtual node I and J after the failure of I while $b_{u,v}^{J,I}$ corresponds to the flow between the virtual node I and J after the failure of J. Thus both variables can be non-zero.

5.3.1 Objective

The objective of the model is to minimize the total consumed resources as shown in Eq. (6.1), where α, β are used to tune the weight of the node and link resources, respective.

$$\min[\alpha \times (\sum_{u,v} r_{u,v} + \sum_{u,v,I,J} w_{u,v}^{I,J}) + \beta \times (\sum_{v} \sum_{U} W_{U,v} \times cr(U) + \sum_{v \in N^S} \gamma_v)] \quad (5.1)$$

5.3.2 Constraints

Virtual Node Mapping For each virtual node, it has to be mapped to *exact one working* and *one backup node* in the substrate network as shown in Eq. (6.2) and (5.3).

$$\sum_{v \in N^S} W_{U,v} = 1 \;\; \forall U \in N^V \tag{5.2}$$

$$\sum_{v \in N^S} B_{U,v} = 1 \;\; \forall U \in N^V \tag{5.3}$$

For each physical network node, we have the constraint in Eq. (6.3) to guarantees that no co-located working nodes, or co-located backup and working nodes.

$$\sum_{Z \in N^V} W_{Z,v} + B_{U,v} \leq 1 \;\; \forall (U, v) \in L^A \tag{5.4}$$

Link Mapping for SNS Problem In the AUG, for a virtual node (e.g., Node *a* in Fig.6.5), the working flow can only be carried on one edge towards the physical nodes (i.e., either Edge *a*-1 or *a*-2), as show in Eq. (6.7), and (5.6).

$$W_{I,v} \times br(I, J) = w^{I,J}_{I,v} \;\; \forall (I, v) \in L^A, (I, J) \in L^V, I < J \tag{5.5}$$

$$W_{J,v} \times br(I, J) = w^{I,J}_{v,J} \;\; \forall (J, v) \in L^A, (I, J) \in L^V, I < J \tag{5.6}$$

We have the flow conservation constraints for the working flow as follows.

$$\sum_{v \in N^S} w^{I,J}_{u,v} - \sum_{v \in N^S} w^{I,J}_{v,u} = 0 \;\; \forall (I, J) \in L^V, I < J, \forall u \in N^S \tag{5.7}$$

$$\sum_{v \in N^S} w^{I,J}_{I,v} - \sum_{v \in N^S} w^{I,J}_{v,I} = br(I, J) \;\; \forall (I, J) \in L^V, I < J \tag{5.8}$$

$$\sum_{v \in N^S} w^{I,J}_{J,v} - \sum_{v \in N^S} w^{I,J}_{v,J} = -br(I, J) \;\; \forall (I, J) \in L^V, I < J \tag{5.9}$$

When the backup flow $b^{I,J}_{u,v}$ is used to survive the failure of I, one has to make sure the flow goes to J via the working node, and departs I via the backup substrate node as shown in Eq. (5.10) and (5.11).

$$B_{I,v} \times br(I,J) = b^{I,J}_{I,v} \;\; \forall (I,v) \in L^A, (I,J) \in L^V \tag{5.10}$$

$$W_{J,v} \times br(I,J) = b^{I,J}_{v,J} \;\; \forall (J,v) \in L^A, (I,J) \in L^V \tag{5.11}$$

We have the flow conservation constraints for the backup flow as follows.

$$\sum_{v \in N^S} b^{I,J}_{u,v} - \sum_{v \in N^S} b^{I,J}_{v,u} = 0 \;\; \forall (I,J) \in L^V, \forall u \in N^S \tag{5.12}$$

$$\sum_{v \in N^S} b^{I,J}_{I,v} - \sum_{v \in N^S} b^{I,J}_{v,I} = br(I,J) \;\; \forall (I,J) \in L^V \tag{5.13}$$

$$\sum_{v \in N^S} b^{I,J}_{J,v} - \sum_{v \in N^S} b^{I,J}_{v,J} = -br(I,J) \;\; \forall (I,J) \in L^V \tag{5.14}$$

Furthermore, we need to guarantee that the augmented edges (e.g., a-1 in Fig. 6.5) carries no backup or active flows other than those originating or terminating at the virtual node a. This is achieved in Eq. (5.15).

$$\sum_{J != I, K != I} w^{J,K}_{I,v} + w^{J,K}_{v,I} + b^{J,K}_{I,v} + b^{J,K}_{v,I} = 0 \;\; \forall (I,v) \in L^A \tag{5.15}$$

Link Mapping for SNN Problem Other than the equations employed in the case for splittable flow, the model of non-splittable flow has to incorporate the following two constraints.

The non-splittable working flow is achieved with Eq. (5.16).

$$x^{I,J}_{u,v} \times br(I,J) = w^{I,J}_{u,v} \;\; \forall (u,v) \in L^S, (I,J) \in L^V, I < J \tag{5.16}$$

The non-splittable backup flow is achieved with Eq. (5.17).

$$y_{u,v}^{I,J} \times br(I,J) = b_{u,v}^{I,J} \ \ \forall (u,v) \in L^S, (I,J) \in L^V \tag{5.17}$$

Node Capacity Constraints The constraints in Eq. (5.18) and Eq. (6.9) guarantees that the allocated node resource is within the available capacity.

$$B_{U,v} \times cr(U) \leq \gamma_v \ \ \forall (U,v) \in L^A \tag{5.18}$$

$$\sum_{U \in N^V} W_{U,v} \times cr(U) + \gamma_v \leq cc(v) \ \ \forall v \in N^S \tag{5.19}$$

Link Capacity Constraints We have Eq. (5.20) to capture the sharing, where $s(I,J)$ gives the right index pair for the flow between (I,J) to follow the convention discussed above. Note that the term $\sum_{J \in N^V} b_{u,v}^{I,J} + \sum_{J \in N^V} b_{v,u}^{I,J}$ represents the total backup flow on an edge when virtual node I fails, which can not be shared among each other. However, above backup flows for the failure of I can share the active flow from I to other virtual nodes (i.e., $\sum_{J \in N^V} w_{u,v}^{s(I,J)} + \sum_{J \in N^V} w_{v,u}^{s(I,J)}$). Hence, the allocated backup resource on Edge (u,v) is the difference of above two terms (when the difference > 0). We note that the sharing between backup flows is also incorporated in the Eq. (5.20). This is because Eq. (5.20) is applied for each virtual node I, while the $r_{u,v}$ in fact takes the maximum (instead of summation) among the backup resources allocated to protect the failure of each virtual node.

$$(\sum_J b_{u,v}^{I,J} + \sum_J b_{v,u}^{I,J}) - (\sum_J w_{u,v}^{s(I,J)} + \sum_J w_{v,u}^{s(I,J)}) \leq r_{u,v} \ \ \forall I \in N^V, (u,v) \in L^S \tag{5.20}$$

Finally, the allocated link capacity of a substrate link should be less than the capacity of the corresponding substrate link, as shown in Eq. (5.21).

$$\sum_{(I,J) \in L^V} w_{u,v}^{I,J} + \sum_{(I,J) \in L^V} w_{v,u}^{I,J} + r_{u,v} \leq bc(u,v) \ \ \forall (u,v) \in L^S \tag{5.21}$$

Above ILP models can obtain optimal solutions for the SNN, and SNS problems, respectively. However, it is also known to be intractable for large networks. To be more computational efficient, we resort to Tabu-search based heuristic algorithms to address SNS, and SNN problems for large instances, which are introduced in the next section. Note that although Tabu-search based approach is theoretically exponential, the termination can be adjusted by tuning the number of search iterations.

5.4 Heuristic Algorithms for the SNF Problem

For larger scale networks, we adopt a two-step approach to achieve the survivability: in the first step, we construct an auxiliary protection graph (APG) [16] that embeds the node protection for the virtual network; in the second step, the APG is mapped to the substrate network where the survivability is transparently employed. In the following, we first discuss the APG construction, present the design for the Tabu-search framework that maps the APG to the substrate network, and then show how to resolve the SNS/SNN problem using above framework, respectively.

5.4.1 Construction of the Auxiliary Protection Graph

To provide survivability against the single facility node failure, one can construct an auxiliary protection graph in the following manner: add an extra backup virtual node; connect the backup node to all the other virtual nodes. This results in an auxiliary protection graph (APG) that embeds the resilience that can survive any single node failure. Figure 5.4 shows the construction of the APG for the virtual network shown in Fig. 6.5. Note that the computing demand of the backup node, say *B* (shown in Fig. 5.4), is the maximum demand among all the other virtual nodes, while the demand for the auxiliary link, e.g., *B*-*a*, is the maximum link bandwidth demand among all the virtual links incident to *a*. With the APG, the SNF problem turns out to be a pure VNE problem, which will be resolved using the Tabu-search method in the next subsection [1].

[1]One reason that motivates us to use Tabu-search is that the virtual node selection process imitates the neighbor exploration process in Tabu-search.

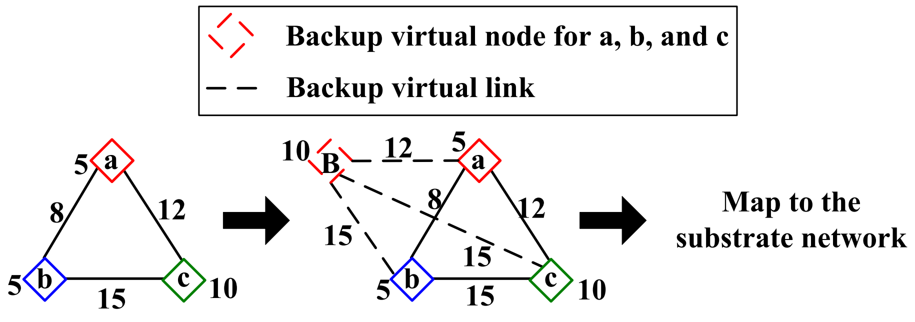

Backup virtual node for a, b, and c
Backup virtual link
5 a
8
12
5 b
15
c 10
10 B
12
5 a
8
15
12
15
5 b
15
c 10
Map to the substrate network

Figure 5.4. Construction of Auxiliary Protection Graph

5.4.2 Virtual Network Embedding based on Tabu-search Meta-heuristic

In the following, we present our design for the Tabu-search framework, including the solution encoding, the neighbor selection and evaluation, and the stagnation avoidance, followed by the overall algorithm.

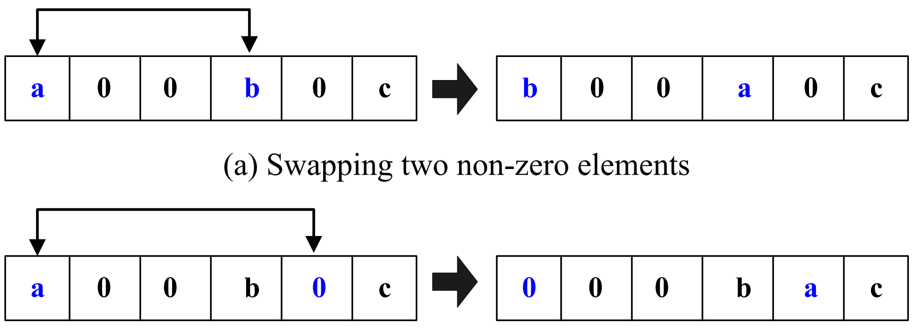
Encoding of the Solution Space for the Node Mapping We denote the set of virtual nodes as $N^V = \{v_1, v_1, ..., v_m\}$ (i.e., $|N^V| = m$), and the set of substrate nodes N^S as $\{V_1, V_1, ..., V_n\}$ (i.e., $|N^S| = n, m \leq n$). The solution of a feasible node mapping can be represented by a vector S of size $n = |N^S|$. For each element s of S, it can have the integer value within $[0, m]$. For the *i-th* element, value 0 represents that no virtual node is mapped to the V_i; while any other value, say j, means that the virtual node v_j is mapped to the substrate node V_i. Figure 5.5 presents the encoding of one feasible solution when mapping three virtual nodes *a*, *b*, *c* onto a substrate network containing six physical nodes *A*, *B*, *C*, *D*, *E*, *F*.

N^V={a, b, c}, N^S={A, B, C, D, E, F}

A	B	C	D	E	F
a	0	0	b	0	c

Figure 5.5. Solution Encoding of the Node Mapping

Neighbor Exploration Given above solution encoding, we introduce two operations that generate the neighbors of the current solution. The first operation is the exchange of two non-zero elements in the solution vector, where the mapping of two virtual nodes are swapped over respective substrates nodes. The other operation is the exchange of one zero element with one non-zero element, which means the virtual node is migrated to the substrate node with the zero element. Note that in both operations, the examination of the node capacity has to be employed to make sure the exchange does not lead to the violation of the substrate node capacity. The example for both operations is shown in Fig. 5.6.



(a) Swapping two non-zero elements

(b) Swapping a non-zero element with a zero element

Figure 5.6. Two Possible Neighbor Solutions Obtained by Swapping

Evaluation of the Neighbor Solution In Tabu-search, a decision has to be made on which neighbor to proceed after evaluating the goodness of each neighbor. In our case, we have to estimate the resource consumption for each neighbor solution since potentially there are exponential paths between each node-pair of the substrate network. Instead of only relying on the shortest path, we use the average hop number of the k-shortest path to approximate the link resource usage.

Rather than calculating the exact resource for each neighbor, we obtain the difference between the current solution and the neighbor solution. We use an example to show this calculation. Assume virtual node a and b is initially mapped onto the substrate node B and C respectively. As a neighbor solution, these two substrate nodes are exchanged (a to C, b to B). We use $N(a), N(b)$

to denote the virtual adjacent nodes of a, b, respectively, and use $M(N(a)), M(N(b))$ to denote the substrate nodes that map the nodes in $N(a), N(b)$. Since the difference on the resource consumption brought by the swapping only applies to the neighbors of a and b, we have the difference calculated with Eq. 5.22. Note that $R(I, J)$ is defined as $H(I, J) \times br(i, j)$, where i, j are the virtual nodes residing on I, J, and $H(I, J)$ is the average hop number for the k-shortest paths between I, J. Since the case for the exchange between one zero and one non-zero element is even easier, we omit the detail here.

$$\sum_{I \in M(N(a))} R(I, B) + \sum_{I \in M(N(b))} R(I, C) - (\sum_{I \in M(N(a))} R(I, C) + \sum_{I \in M(N(b))} R(I, B)) \tag{5.22}$$

Stagnation Avoidance To prevent the searching process from cycling in a small set of solutions, we also maintain both the recency and the frequency information for the exchanged pair to enable a short-term control and long-term control, respectively. Basically, the recently exchanged pair is stored in the tabu recency list to make sure no reverse exchanges in the near future. To allow the searching process to explore other parts of the solution space, we store the global number of exchanges of each substrate node pair in the frequency list. The pair with the least frequency value will be chosen for exchange when no promising neighbors exist or no improvement has been observed for a while.

When combining above aspects together the overall process is shown in Algorithm 3. Where the variable Cur_Sol, and $Best\_So\_Far$ represent the current solution, and the best solution found so far, respectively. Also note that we don't directly accept a neighbor solution in Line 9. Instead, we have to check the feasibility of the resulted Multi Commodity Flow (MCF) problem in Line 16 before updating the solution. The main difference of the SNS and the SNN problem in fact lies on the complexity of the resulted MCF problem, which is further elaborated below.

5.4.3 Resolve the MCF for SNS and SNN Problems

We present the detailed procedure to resolve the MCF for SNS, and SNN problems in this subsection.

Algorithm 4 Tabu Search Algorithm for Virtual Network Embedding

```
 1: Generate a initial feasible solution as the $Cur\_Sol$, and use the resulted resource amount as
    $Best\_So\_Far$;
 2: $i \leftarrow 1$
 3: while $i \leq No\_Of\_Iter$ do
 4:     BEGIN: Find the best neighbor solution of $Cur\_Sol$, namely $Best\_Nbr$;
 5:     if $Best\_Nbr$ is better than $Best\_So\_Far$ then
 6:         Set the temporary solution ($Temp\_Sol$) as $Best\_Nbr$;
 7:     else
 8:         if There exists neighbors better than $Cur\_Sol$, and not in the Tabu recency list then
 9:             Set the $Temp\_Sol$ as the best one among those neighbors;
10:         end if
11:     else
12:         if No neighbor is better then
13:             Choose the least frequent solution in the Tabu frequency list as the $Temp\_Sol$;
14:         end if
15:     end if
16:     Solve the resulted MCF problem;
17:     if No feasible optimal solution found based on $Temp\_Sol$ then
18:         Mark infeasibility of the chosen solution, and go back to BEGIN
19:     end if
20:     Set the $Cur\_Sol$ as $Temp\_Sol$, update the Frequency and Tabu list, and $Best\_So\_Far$;
21:     $i \leftarrow i + 1$
22: end while
```

SNS MCF Problem For the case with splittable flow, the resulted MCF problem can be reduced to a linear programming problem. Theoretically, it hence can be resolved in polynomial time using the ellipsoid algorithm [36]. In practice, however, we can simply rely on a lagrange-relaxation solution or column-generation solution to obtain the optimal result. Since both solutions are standard approaches for the MCF problem [42], we omit the detail here.

SNN MCF Problem For the case with non-splittable flow, the resulted MCF problem restricts the flow to be carried by only one path, which is NP-hard [47] [48]. In this study, we adopt the following relaxation process to resolve this problem.

1. Resolve the LP-Relaxation of the non-splittable flow MCF problem. For each commodity, the solution consists of one or multiple active paths.

2. Start with the commodity that has the minimum number of active paths, choose the active path that carries the maximum flow.

3. If the chosen active path leads to the violation of the link bandwidth bound, choose the next active path.

4. Continue this process until all the commodity select one path for carrying the flow.

5.5 Performance Evaluation and Analysis

In this section, we evaluate the proposed schemes and compare the performance. Both the virtual network and substrate network are randomly generated with the number of nodes falling in a given range. And the virtual/substrate links are randomly decided for each node pair that leads to an average network connectivity of 50% (i.e., 50% existing possibility of the link between any node pair). We set $\alpha = \beta = 1$ in the objective of the ILP model. For large scale networks, ILP model is time-inefficient (especially in a dynamic traffic context). We hence compare the performance of ILP models and the heuristics on small networks and obtain the average blocking probability of VN requests for all approaches. In addition, we compare other metrics, including resource consumption, QoS metrics, and the impact of the simulation parameters in the second subsection with a different setting.

5.5.1 An Exemplary Comparison of SNN and SNS

To better understand the difference between splitable and non-splitable flow mapping, We first compare the resource usage of SNS and SNN through an example based on the virtual network and substrate network shown in Fig. 6.5. The solutions are obtained through the ILP model.

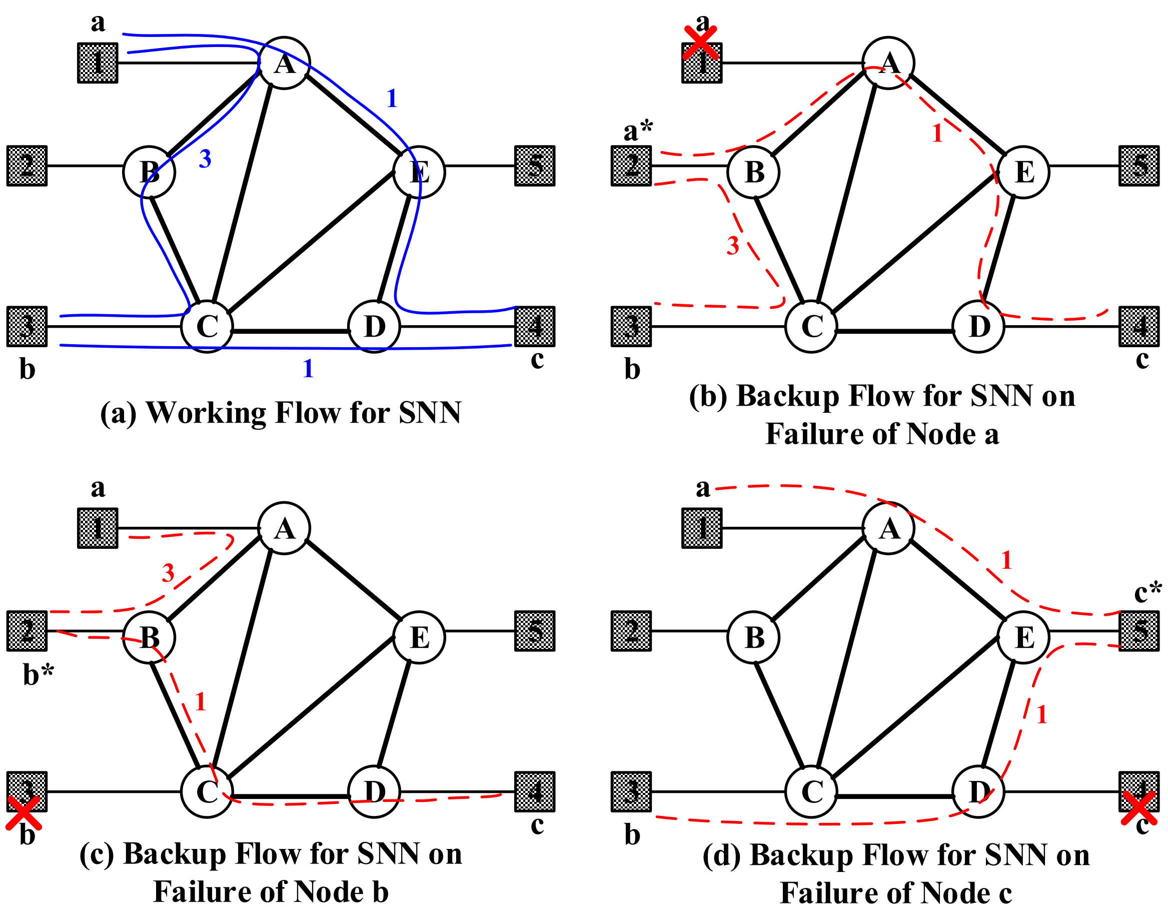

a
1
A
1
3
2
B
E
5
3
C
D
4
b
1
c
(a) Working Flow for SNN
a
1
A
a*
1
2
B
E
5
3
3
C
D
4
b
c
(b) Backup Flow for SNN on Failure of Node a
a
1
A
3
2
B
E
5
b*
1
3
C
D
4
b
c
(c) Backup Flow for SNN on Failure of Node b
a
1
A
1
c*
2
B
E
5
1
3
C
D
4
b
c
(d) Backup Flow for SNN on Failure of Node c

Figure 5.7. Example Solution with SNN

The non-splittable flow ILP solution is shown in Fig. 5.7. For each virtual node (say *a*), the corresponding mapped working substrate node is labeled as *a*, and the backup node is labeled as

$a*$. To present a clear picture of the whole solution, we separate the flows into four parts. Fig. 5.7(a) shows the working flows of the solution, the number beside each flow is the value of the flow. Fig. 5.7 (b) illustrates the backup flows upon the failure of node a. Fig. 5.7 (c) shows the backup flow upon the failure of node b. Fig. 5.7 (d) demonstrates the backup flows upon the failure of node c. In this case, the objective of the optimal solution is 44.

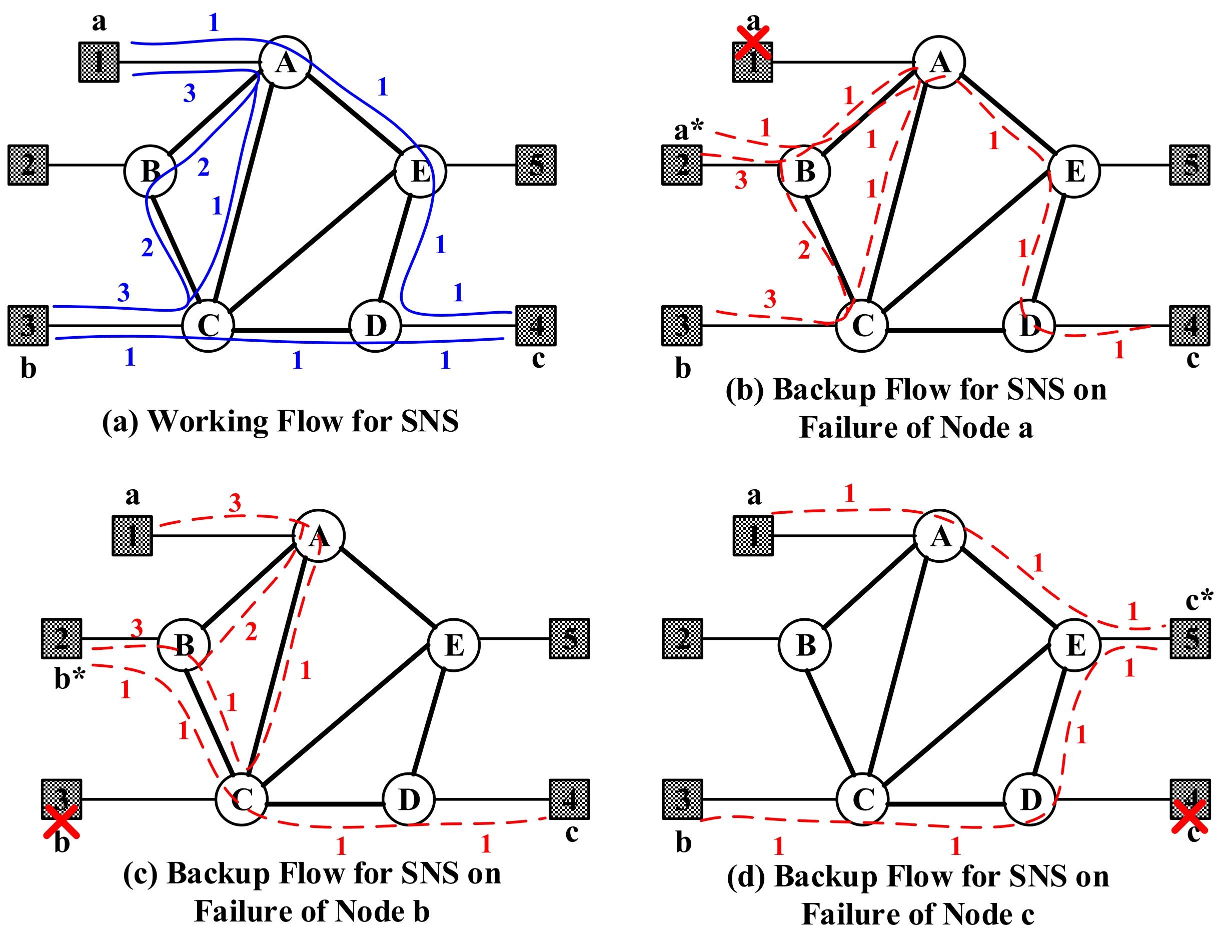

(a) Working Flow for SNS
(b) Backup Flow for SNS on Failure of Node a
(c) Backup Flow for SNS on Failure of Node b
(d) Backup Flow for SNS on Failure of Node c
a
b
c
a*
b*
c*
A
B
C
D
E
1
2
3
4
5

Figure 5.8. Example Solution with SNS

Comparatively, Fig. 5.8 shows the ILP solution for splittable flow mapping. Similarly, Fig. 5.8(a) shows the working flow of the solution. Note that the virtual link a-b of 3 units of bandwidth requirement has been satisfied by splitting it into a flow of 2 units and a flow of 1 unit. Each number represents the amount of flow carried by the corresponding link for a specific flow. Fig.

5.8(b) shows the backup flow upon the failure of node *a*. Fig. 5.8 (c) illustrates the backup flows upon the failure of node b. Fig. 5.8(d) shows the backup flow upon the failure of node c. Note that the flow splitting also applies to the backup flow when node *b* and a fail. In this case, the objective of the optimal solution is 43, which is smaller than 44.

5.5.2 Comparison of ILP Models and Heuristic Algorithms: Small Networks

In this subsection, we compare the ILP models and the heuristic algorithms in terms of the blocking probability ($\frac{Blocked\ Requests}{Total\ Requests}$) for a given period (i.e., $10,000$ time units). We assume that VN requests arrive in a Poisson process with a rate of r VNs per 10 time units, and stay with 100 time units following an exponential distribution. The number of nodes for the virtual network is randomly generated within $[2,4]$, and the size of the substrate network is fixed to be 10. The bandwidth/computational requirement of the virtual network link/node is randomly generated within $[1,5]$, while the link/node capacity of the substrate network is randomly generated within $[10,50]$.

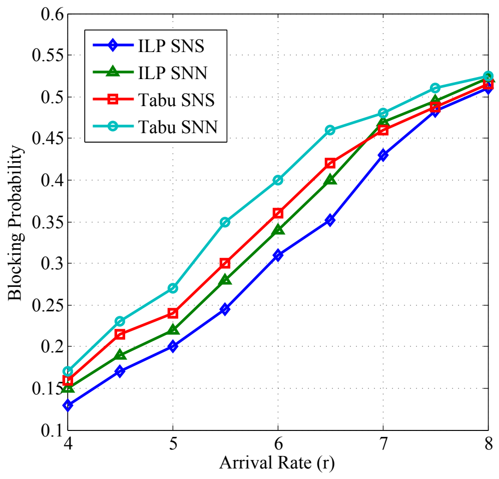


Figure 5.9. Comparison of Blocking Probability among Different Methods

The comparison of the blocking probability is shown in Fig. 5.9. Clearly, the increase of the arrival rate leads to the growth of the blocking probability, and ILP-based solutions outperform the respective heuristics in most case. However, with large arrival rate, the ILP-based solutions have no evident advantage over the respective Tabu-based heuristics. This is due to the fact that greedily (though optimal for the present network state) allocating the resources in ILP models for the current VN request does not necessary lead to better acceptance chance for future requests. In addition, we observe lower blocking probability in the case of splittable mapping due to the increased path search space in the link mapping. The difference between the case with splittable and non-splittable mapping is especially evident when the network load is high due to the faster saturation of network bandwidth. Particularly, when the arrival rate is high, we observe cases that the heuristic algorithm with splittable mapping even outperforms the ILP-based solution with non-splittable mapping, which demonstrates the advantage of the former.

5.5.3 Comparison of ILP Models and Heuristic Algorithms: Large Networks

In this subsection, we compare ILP models with the heuristic algorithms in terms of the resource consumption, the QoS metric, and study the impact of the iteration number in the heuristics. The number of nodes for the virtual network request is randomly generated within $[2, 10]$, and the size of the substrate network falls in the range of $[10, 50]$. The bandwidth/computational requirement of the virtual network link/node is randomly generated within $[1, 20]$, while the link/node capacity of the substrate network is randomly generated within $[1, 50]$. We generate multiple thousands of the instances and report the average performance (i.e., QoS, resource consumption) below after obtaining the performance of each running instance.

Comparison of Resource Consumption The resource comparison between different approaches is presented in Fig. 5.10, where the X-axle is the number of nodes for the virtual network, and the Y-axle shows the total consumed resources for respective schemes, where the total consumed resources refers to the node/link resources consumed for both active and backup traffic. From Fig. 5.10, we can have three major observations. First, the schemes with splittable flow clearly outperform the schemes with non-splittable flow. Second, the Tabu-search based solution for the SNS problem leads to less consumed resources, when compared to the optimal results for the SNN problem. Both of these observations are due to the fact that the splittable flow enables a larger solution (i.e., path) space for the virtual link mapping. Third, the proposed Tabu-search

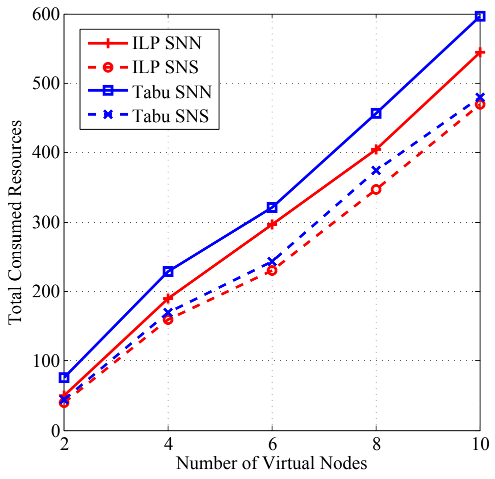


Figure 5.10. Resource Comparison among Different Methods

schemes perform very close to the optimal results from the respective ILP models. Considering the extensive computational overhead of the ILP models, in practice, the Tabu-search based scheme thus can be an effective alternative solution. The Tabu-search scheme above is configured to run 1000 iterations. In the next, we will further study the impact of this parameter.

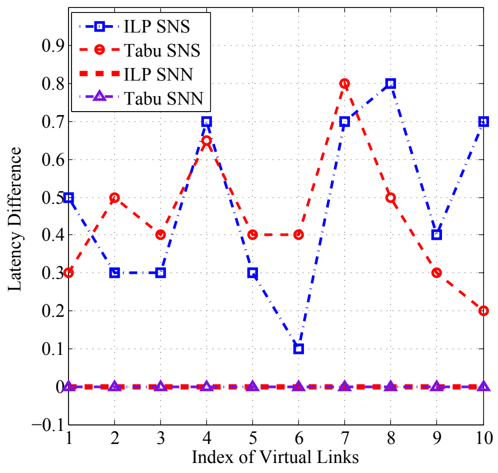


Figure 5.11. QoS Comparison among Different Methods

QoS Comparison of the ILP Models and Heuristic Algorithms The splittable flow leads to better resource performance. However, it may bring the tradeoff in the QoS. In our evaluation, to see the impact, we have randomly generated the latency of each substrate link within $[0.1, 1]$. For each commodity (i.e., a virtual link), we calculate the difference between the active path with the maximum latency and the one with the minimum latency as $\frac{Max-Min}{Max}$. The average result for a virtual network with 10 virtual links are presented in Fig. 5.11, where the X-axle is the index of the virtual link, and Y-axle shows the path latency difference. Clearly, for the non-splittable flow mapping, the difference stays at 0 since only a single path is chosen. For the schemes with splittable flow, instead, we observe an evident difference for most virtual links [2]. In practice, packets (along the same virtual link) hence may go through different substrate paths and experience various delay in the splittable mapping, resulting in issues such as packet disorder. As a result, how to employ splittable mapping without significant latency delay, has to be well studied, which can be addressed in a path-based formulation [49] that can restrict the latency difference among selected paths.

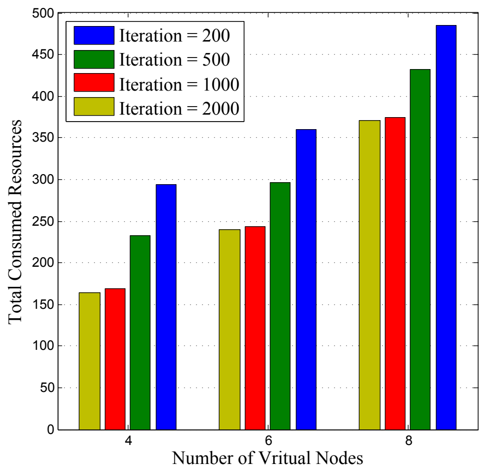


Figure 5.12. The Impact of Iteration Number

[2]Results based on varying the latency range for each link or other parameters show the similar observation that most virtual links posses a non-neglectable latency difference, thus omitted here.

The Impact of the Tabu-search Parameter One advantage of Tabu-search approach over the ILP model is the time efficiency and its flexibility in adjusting the running time by adopting different iteration number in Algorithm 1. We show the impact of the number of iterations in Fig. 5.12. In Fig. 5.12, the X-axle, Y-axle again shows the number of nodes for the virtual network, and the total consumed resources, respectively. With varying the iteration number, one can clearly observe a resource reduction. When the iteration number is more than 1000, the improvement, however, is not as evident. As a result, in practice, one may adopt a medium-size iteration number, bewaring of the time overhead with larger iteration number.

5.6 Remarks

In this chapter, viewed from a network flow perspective, the SNF problem is modeled as ILP formulations which considers joint working and backup requests for both splittable and non-splittable flow mappings. For larger-scale problems, Tabu-search based heuristic algorithms are proposed to provide a practical and efficient solution. We also evaluate the correctness and efficiency of the proposed ILP models and heuristic algorithms and present our results. In the next step, we plan to study other failure scenarios including, for instance, two failures or multiple failures of facility nodes.

PART 6

OPTICAL NETWORK VIRTUALIZATION

The Internet traffic has been growing at a dramatic speed, which almost doubles every five years [28]. At the same time, the power consumption of the current Internet has brought significant concern regarding the economic, energy, as well as environmental implication of the current Internet. Fortunately, optical networks promises to address these issues given the well-known high bandwidth and energy efficiency [18] [19].

An virtualized future Internet apparently calls for the support of bandwidth-abundant optical-based substrate networks. To fully reap the advantage of optical networking, it is ideal to virtualize the optical resources to support any-to-any bandwidth-abundant connectivity for the service layer in network virtualization. Via abstraction, partition, and aggregation of optical resources (e.g., optical link, port, and switch node as shown in Fig. 6.2), optical network and switching allows for more agility and flexibility to satisfy the service request from the users, resulting in a *Connectivity as a Service (CaaS)* to the SPs.

However, optical resources are heterogeneous and optical transmission bear numerous physical limitation due to the analogy feature of optical signal. These brings in new challenges to network virtualization over an optical-based substrate. In this chapter, we present a holistic view of the motivations, architecture, and challenges on the way towards a virtualized Internet supported by optical substrate networks. Among all, we focus on the added dimension of complexity to the VNE problem and present an optimization model for the VNE problem over a virtualized *wavelength-division multiplexing* (WDM) network. To call for more actions on this direction, we also highlight a few open problems that deserve further study. Different from prior work, this chapter focuses on the *integration* of optical virtualization and network virtualization via enabling optical *Connectivity as a Service*.

The remaining of this paper is organized as follows. Section 6.1 discusses the motivations and

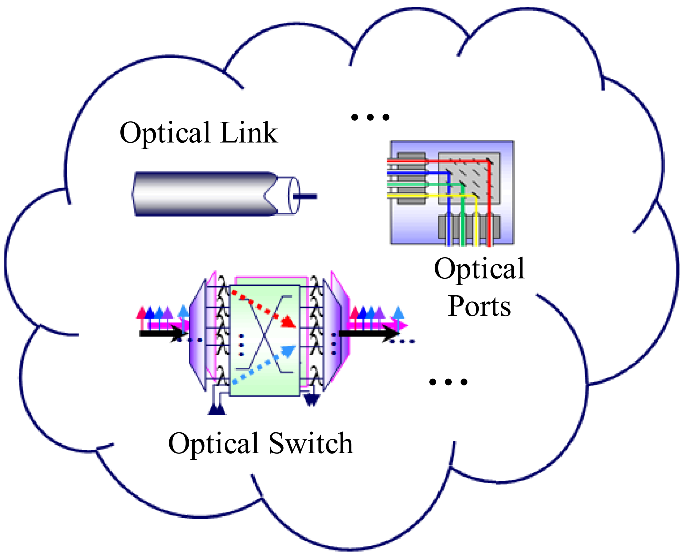


Figure 6.1. Optical Virtualization

challenges of optical-based network virtualization. In Section 6.2, we present the architecture to enable network virtualization upon a virtualized optical substrate. We formally define the optical VNE problem in Section 6.3, and present an optimization model for the WDM VNE problem in Section 6.4. Open problems and further discussions are presented in Section 6.5. Finally, we conclude this chapter in Section 6.6.

6.1 Optical-based Network Virtualization: Motivations and Challenges

In this section, we discuss the major motivations behind network virtualization, optical virtualization, and the incentive/challenges for the optical-based network virtualization.

6.1.1 Drive of Network Virtualization and Optical Virtualization

As we have extensively discussed in early chapters, the major drive for network virtualization is to overcome the impasse of the current Internet [2] [3] [4] [5] where disruptive technologies can barely be employed due to multiple architecture constraints of the present Internet Infrastructure. For instance, the stakeholder is reluctant to adopt revolutionary technologies due to the potential CAPEX/OPEX cost. Also, with current Internet, a global agreement and coordination are needed for employment of new technologies, which, however, is hard to achieve in practice. Those barriers are removed with the idea of creating a layer of abstraction by decoupling the traditional ISPs into the Infrastructure Providers (InPs) (which manage the physical infrastructure), and the Service Provider (SPs) (which manage the virtual network and offer services to the end users) [2] [3] [4] [5]. With this decoupling, more flexibility and convenient management are allowed since SPs can deploy services without high investments on the physical infrastructure while InPs can operate and upgrade without affecting supported services.

The IT industry is experiencing an era of virtualization where software, platform, infrastructure, (or anything), are all abstracted and virtualized as services, enabling a *pay-as-you-go* business model. The major driver of this revolution is the resulted cost savings, management overhead reduction, as well as increased adaptability. Among this wave, it is not surprising that the need for virtualized optical networks is raised recently, and attracts significant interests. The idea of optical virtualization was initially introduced in [29]. It is shown in [29] that when automated optical path provisioning, segmentation/aggregation of network resources, and optical network resource management/coordination are all feasible, optical networks can be virtualized to provide stronger and smarter cooperation to the upper layer. The authors of [30], from a practical viewpoint, discussed major optical enabling technologies, devices, architectures. Specifically, the resource partition and aggregation for various optical resources including switch ports and link capacity are all investigated. With a virtualized optical network, various applications benefit from the huge optical bandwidth, including data center and cloud computing applications, which is discussed in [31]. Recently, the authors of [32] studied the RWA problem in a virtualization context. Different from above work, in this paper, we introduce the concept of *Connectivity as a Service*, and investigate the *integration* of above two trends.

Note that there are multiple advantages with a virtualized optical network. First, it introduces the flexibility of provisioning high-bandwidth any-to-any connectivity. Second, it enables the agility of customizing the bandwidth provision to adapt the change of the user requests. In addition, a virtualized optical networks can enable more time-multiplexing of the deployed optical resources (in a timely and automated manner), resulting in more revenues to the stakeholder.

6.1.2 Incentive for Optical-based Network Virtualization

The Internet traffic explosion is expected to be continued in the foreseeable future [28]. Optical networking is considered as the foremost solution to handle this ever-increasing bandwidth demand. Therefore, it is desirable that a virtualized next generation Internet is build upon a virtualized optical substrate. To be seamlessly integrated into network virtualization where the user requests can work in a *pay-as-you-go* fashion, the optical resources have to be virtualized to enable *connectivity as a service* for the upper layer. We refer to the resulted *integration* of network virtualization and optical virtualization as *Optical-based Network Virtualization*.

6.1.3 Challenges in Optical-based Network Virtualization

While virtualized optical networks promise a highly flexible, and bandwidth-abundant substrate for network virtualization, this integration faces multiple important challenges [1].

First, due to the analog nature of optical transmission, optical signal is prone to degradation and impairment. The availability of an optical connection is impacted by the signal reachability (which is also decided by the modulation level). When signal regeneration is needed, the availability of a connection is determined by the availability of regeneration devices.

Second, optical spectrum is organized in granularity of wavelength (or sub-carriers) in WDM networks (or OFDM-based optical networks). In contrast, in a general (packet-switched) network virtualization context, dimensionless bandwidth resources are assumed. For sub-wavelength and super-wavelength traffic, the wavelength (or sub-carrier) resources need efficient partition and aggregation, leading to the challenges to the network control plane. In addition, the employed wavelength (or sub-carrier) for a request has to follow the *wavelength continuity* constraint (in the absence of wavelength converters), which further complicates the VNE process.

Third, virtualized optical networks are expected to be remotely controllable for flexible and agile provision. This indicates that all the resources (with aggregation and partition) associated with the connection should be accessible by the control plane, which potentially demands for ex-

[1]Note that we omit the challenges under the category of the classic network virtualization, which have been extensively discussed in other work (e.g., [1]).

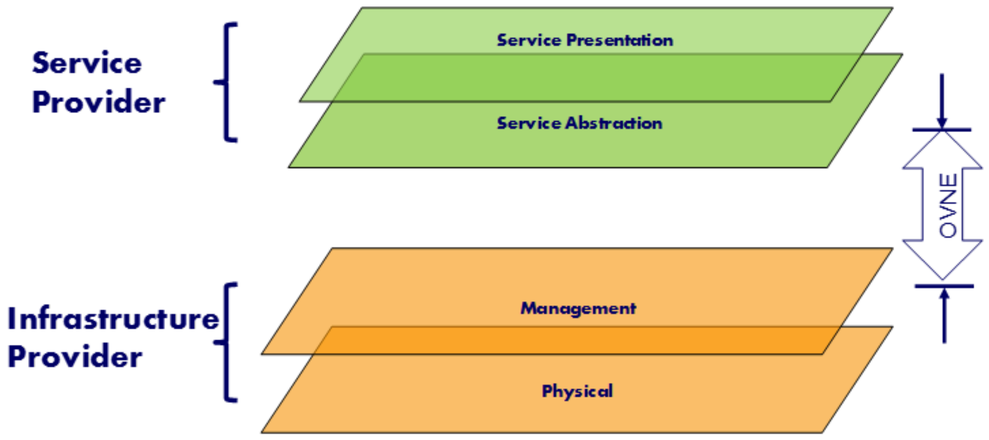


Figure 6.2. Optical-based Network Virtualization

tending the GMPLS control plane and/or using the Software Defined Networks (e.g., the OpenFlow architect [50]).

Fourth, from the algorithmic viewpoint, the classic *Virtual Network Embedding* problem is already an NP-Complete problem [8]. The optical-related constraints add extra dimensions of complexity to the VNE problem.

Furthermore, optical networks diversify in terms of the switching paradigms (optical burst/-packet/circuit switching [51] [52]), node architectures, spectrum management policy (i.e, WDM networks or OFDM networks). We exclude all those complexities, and provide a generic architecture for network virtualization over the optical substrate in the next section.

6.2 Optical-based Network Virtualization: Architecture Overview

Figure 6.2 shows the overall architecture for optical-based network virtualization architecture in our vision, which consists of four major layers, where the top two layers belongs to the SPs and the bottom two layers belongs to the InPs.

First, at *Service Presentation* layer, user can interact and customize their requests (via, for example, a HTTP REST API interface) based on their needs, and budgets. The *Service Abstraction* layer, on the other hand, abstracts the user requests as logical *virtual networks*. The detailed components of these two layers are shown in Fig. 6.3. An example virtual network is shown in

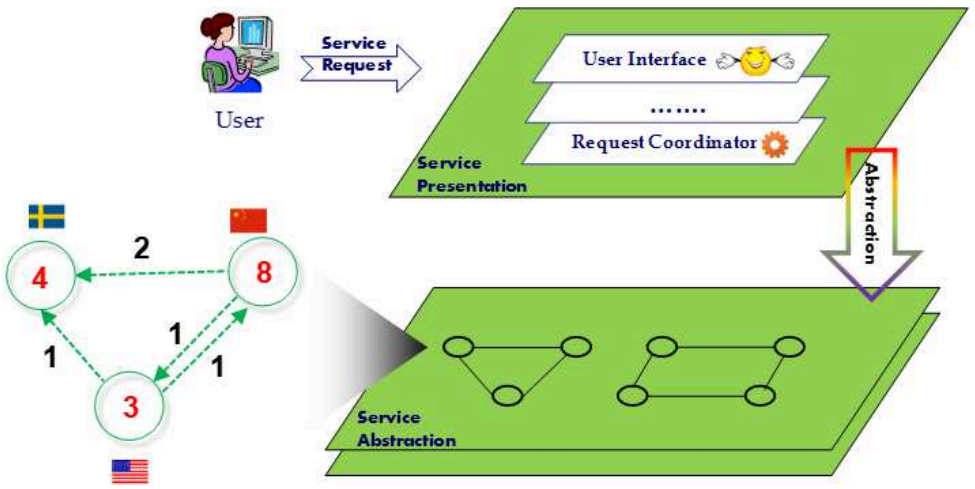


Figure 6.3. Optical-based Network Virtualization: Layers of SPs

the bottom layer of Fig. 6.3 where three facility nodes (i.e., nodes support computing or storage capabilities) are connected with four direction lightpaths where the number besides each link represents the needed number of lightpath according to the bandwidth requirement between incident facility nodes.

The next two layers are detailed in Fig. 6.4. At The *Physical Control and Management* layer, the optical resources are abstracted and managed by the optical resource manger, and *Connectivity as a service (CaaS)* is offered. Likewise, the utilities resources (e.g., computing services, storage devices) are managed by utility resource manager, *Utility as a service* can be provided. Both types of resources are orchestrated together to offer *Infrastructure as a service* to the service layer (via the VNE process). Finally, the *Physical* layer consists of the physical servers/devices that are connected by optical networks (which consists of optical switch nodes, and optical fiber links).

6.3 Optical Virtual Network Embedding (OVNE)

In this section, we present a generic network model for the most important piece of above architecture: the Optical Virtual Network Embedding (OVNE) problem.

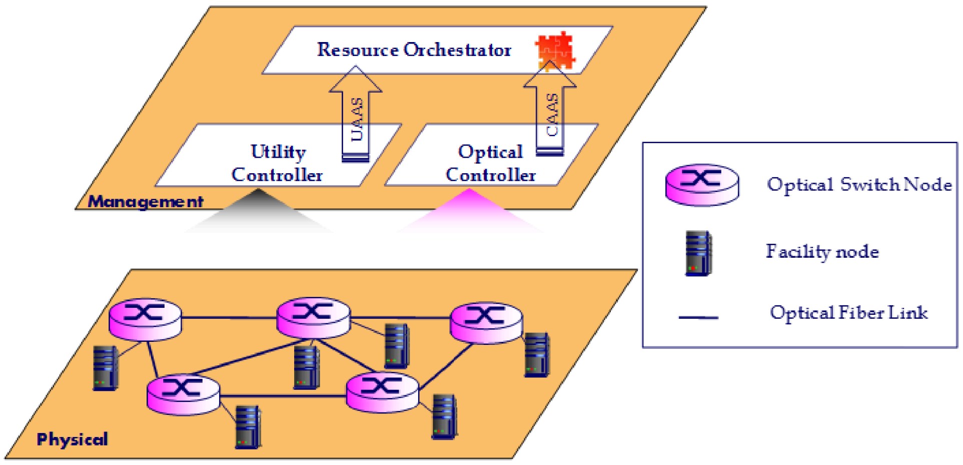


Figure 6.4. Optical-based Network Virtualization: Layers of InPs

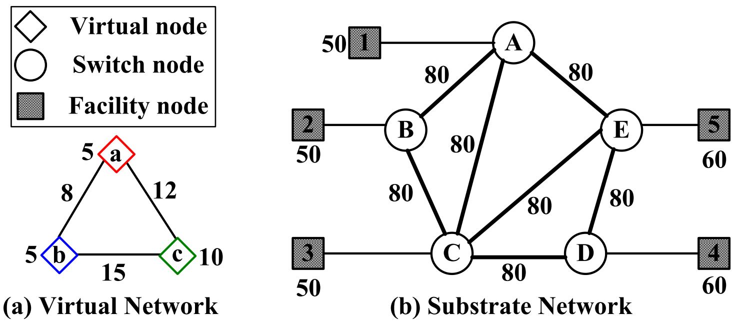


Figure 6.5. Virtual Network and Substrate Network Model

6.3.1 Network Model for OVNE Problem

The virtual network request is represented as a directed weighted graph [2] $G^V = (N^V, L^V)$, where N^V is the set of virtual nodes, and L^V is the set of virtual links. The utility resource requirement of a virtual node a is denoted as $cr(a)$, while the communication demand of a virtual link (between virtual node a and b) reflects the bandwidth requirement between the two incident nodes (i.e., a and b) of the link, denoted by $br(a, b)$. Since it is a directed graph, note that $br(a, b)$ may not be equal to $br(b, a)$. An example of virtual network is shown in Fig. 6.5(a) where there are three virtual nodes, connected by four virtual links, and the numbers besides the nodes/links represent the respective utility resource and bandwidth requirements.

[2]Different from VNE problem defined early, here we assume that the virtual link (i.e., lightpath(s)) is directional.

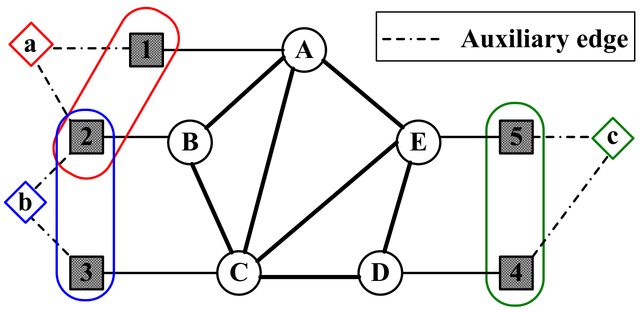


Figure 6.6. Auxiliary Graph for Integrated Virtual Network and Substrate Network

The substrate network is similarly represented by a directed weighted graph $G^S = (N^S, L^S)$ (see Fig. 6.5(b) for an example), where L^S is the set of fiber links (with one fiber per direction for each link), and N^S consists of two types of nodes: the *facility* nodes (denoted as FN) and the optical *switch* nodes (denoted as SN). The former is provisioned with the utility capability and hosts the virtual node, and the latter connects facility nodes to form the substrate network. We denote the available capacity of facility node I as $cc(I)$, and the available bandwidth (e.g., number of wavelengths) of a substrate link a-b as $bc(a, b)$. For the example substrate network shown in Fig. 6.5(b), we have five facilities nodes, connected by five optical switch nodes, where the numbers besides the facility nodes/links represent the available utility and bandwidth resources, respectively. To facilitate the modeling, we further construct an auxiliary graph (AUG) that integrates the substrate network and the virtual network. In specific, for each virtual node (e.g., Node a in Fig. 6.5(a)), we add auxiliary edges to connect it to a group of substrate network facility nodes (e.g., Node $1, 2$ for virtual node a in Fig. 6.6) which satisfy the node capacity constraint (i.e., $cr(a) \leq cc(1)$, and $cr(a) \leq cc(2)$) as well as other constraints such as the location requirement [12].

6.3.2 Definition for OVNE

The Optical Virtual Network Embedding (OVNE) problem can be formally defined as follows.

Definition 3: **Optical VNE Problem.** Given the virtual network $G^V = (N^V, L^V)$, and optical substrate network $G^S = (N^S, L^S)$, the OVNE problem is a decision problem that determines whether it can map the virtual network to the substrate network satisfying the following constraints:

(i) for each virtual node, it is mapped to the substrate network meeting the capacity constraint; *(ii)* for each virtual link, it is mapped to the substrate network meeting the capacity constraint; *(iii)* all the optical-related constraints depending on the specific optical network (e.g., *wavelength continuity* constraint in WDM networks) are all satisfied.

6.4 Mathematic Model of the WDM OVNE Problem

In this section, we present a mathematical model for the optimization version of the OVNE problem over a virtualized WDM network. We assume that there is no wavelength converters and the optical signal can cover the whole physical network. The proposed model is based on the auxiliary graph mentioned above and employs the following variables or notations.

$f_{u,v}^{s,d,w}$:	1 if a lightpath from virtual node s to virtual node d using wavelength w, going through link (u, v) (of the auxiliary graph), 0 otherwise;
$M_{u,v}$:	1 if virtual node u is mapped on to the physical node v, 0 otherwise;
A_v:	available resources on facility node v;
r_u:	utility requirements of virtual node u;
B:	a big integer number;
W:	the number of wavelengths per fiber;
FN:	the set of facility nodes;
$R_{s,d}$:	bandwidth requirements (in terms of number of lightpaths) of virtual link (s, d).

6.4.1 Objective

The objective of the model is shown in Eq. (6.1), which is the summation of all the used wavelength resources over all the fiber links. Moreover, the node resources are excluded in the objective since the consumed node resources are fixed under any mapping.

$$\min[\sum_{w\in[1,W]}\sum_{(s,d)\in L^V}\sum_{(u,v)\in L^S} f_{u,v}^{s,d,w}] \tag{6.1}$$

6.4.2 Constraints

Virtual Node Mapping For each virtual node, it has to be mapped to *exactly one facility node* in the substrate network. This is reflected in Eq. (6.2).

$$\sum_{v:(u,v)\in AUG} M_{u,v} = 1 \ \ \forall u \in N^V \tag{6.2}$$

Meanwhile, for each facility node, we have the constraint in Eq. (6.3), which guarantees that no two different virtual nodes are mapped to the same physical node.

$$\sum_{u:u\in N^V,(u,v)\in AUG} M_{u,v} \leq 1 \ \ \forall v \in FN \tag{6.3}$$

Virtual Link Mapping The mapping of a virtual link is realized as lightpath(s) over the auxiliary graph. We have the following flow conservation constraints for source, destination, and intermediate node, respectively. Note that the *wavelength continuity* constraint is ensured with Eq. (6.4) since it is applied to each wavelength w.

$$\sum_{u:(u,v)\in L^S} f_{u,v}^{s,d,w} - \sum_{x:(v,x)\in L^S} f_{v,x}^{s,d,w} = 0 \ \ \forall(s,d) \in L^V, \forall v \in N^S, \forall w \in [1,W] \tag{6.4}$$

$$\sum_{v:(s,v)\in AUG}\sum_{w\in[1,W]} f_{s,v}^{s,d,w} = R_{s,d} \ \ \forall(s,d) \in L^V \tag{6.5}$$

$$\sum_{u:(u,d)\in AUG}\sum_{w\in[1,W]} f_{u,d}^{s,d,w} = R_{s,d} \ \ \forall(s,d) \in L^V \tag{6.6}$$

In the auxiliary graph, for a virtual node, the flow can only be carried on the edge towards the facility node that maps it (i.e., $M_{u,v} = 1$). We need the following constraint in Eq. (6.7) to guarantee that. In specific, when $M_{u,v} = 0$ (i.e., virtual node u is not mapped to v), Eq. (6.7) ensures that no wavelength can be assigned on the auxiliary edge (u, v). When $M_{u,v} = 1$, the large number B makes sure that Eq. (6.7) is an tautology and wavelengths are allowed to be used on edge (u, v).

$$\sum_{w\in[1,W]} \Big(\sum_{d:(u,d)\in L^V} f_{u,v}^{u,d,w} + \sum_{s:(s,u)\in L^V} f_{v,u}^{s,u,w} \Big) \le M_{u,v} \times B \;\; \forall (u,v) \in AUG, u \in N^V, v \in N^S \tag{6.7}$$

Due to the construction, the auxiliary edge (say u-v that connects virtual node u and facility node v) cannot carry lightpaths that are not originated or terminated at u, which is achieved by Eq. (6.8).

$$\sum_{w\in[1,W]} \sum_{(s,d)\in L^V, s\neq u, d\neq u} (f_{u,v}^{s,d,w} + f_{v,u}^{s,d,w}) \le 0 \;\; \forall (u,v) \in AUG, u \in N^V, v \in N^S \tag{6.8}$$

Node Capacity Constraint For a facility node (say v), the total allocated capacity should be less than its capacity This is reflected in Eq. (6.9).

$$\sum_{u:u\in N^V,(u,v)\in AUG} (r_u \times M_{u,v}) \le A_v \;\; \forall v \in FN \tag{6.9}$$

6.5 Discussions and Open Problems

To enable *Connectivity as a Service*, optical resources should be remotely manageable and automatically provisionable. This automation and remote control largely depend on the proper *abstraction* of optical link resources and node resources [3] to allow easy access by the control plane. Along this direction, more studies/experiments/testbeds are expected to be built to compare/validate proposals from the literature (e.g, [30]).

The model presented in Section 6.4 assumed a WDM-based optical substrate, which can be extended and adapted to fit different settings (e.g., the case with the presence of wavelength conversion, and the case that reachability has to be considered). Also, recent studies demonstrate

[3] See [30] for a detailed discussion on the optical node virtualization.

that a virtualized OFDM-based optical networks have a great potential to better serve the future Internet [53] [29] [54]. This is due to the improved signal quality, and the fine granularity for traffic accommodation in OFDM-based networks. Note that, however, OFDM-based optical networks bear more complexities due to the *spectrum consecutiveness* constraint, as well as the freedom of modulation selection. These add extra difficulties to the OVNE problem, which require intelligent algorithm/protocol design.

It is also important to investigate how the *multicast* pattern can be realized in network virtualization, and particularly, over an optical substrate. Traditionally, optical multicast is achieved via the construction of light-trees [55], which start at the source node of a multicast request and split the signals at intermediate nodes to reach all the destination nodes. The signal splitting capability is supported with the optical splitter and delivery switches, which are also referred as multicast capable optical cross-connects (OXCs). In network virtualization, the user may not designate the source and destination node sets to the service provider, which, instead, are decided in the VNE process. Also, the multicast may be achieved without multicast capable optical OXCs [56], or with limited support by employing tap-and-continue switches [57]. All those variations pose new challenges to the multicast over an optical-based network virtualization, and deserve further study.

In addition, in an optical-based network virtualization context, heterogenous types of network entities including, for example, optical fibers/nodes, utility servers/devices coexist. The failure of those entities could impact the upper layer to a different degree, and requests robust proactive/active protection schemes. In other words, more research dedicated to the survivability should bring the optical features as well as traditional optical survivability efforts (e.g., the *SRLG* constraint [58], the *p-cycle* protection schemes [59]) into account.

6.6 Remarks

Optical virtualization is an important step to support a virtualized next-generation Internet. With the huge bandwidth inherently provided by optical transmission, optical virtualization can supply *Connectivity as a Service (CaaS)* by providing virtualized any-to-any bandwidth-abundant connections to upper layers. This chapter has presented a holistic view on the motivations, architecture, and challenges in optical virtualization, and studied the *optical virtual network embedding* (OVNE) problem. We have presented an optimization model for the OVNE problem over a virtualized WDM network. In the next step, we expect to address the challenges and open problems elaborated in this work, and extend the proposed model to more comprehensive cases.

PART 7

CONCLUSIONS AND FUTURE WORK

In this chapter, we briefly summarize the overall work of this dissertation, and point out a few areas of future study.

7.1 Summary of the Dissertation

The role decoupling of the infrastructure provider (InP) and the service provider (SP) enables a virtualized view of the underlying network infrastructure to the SPs as well as the freedom of adopting the technological advancement to both. This decoupling, at the same time, demands for solving the virtual network embedding problem.

Given the NP-Completeness of the VNE problem, in this dissertation, our first study presents an optimal path-based Integer Linear Programming VNE model. Based on the model, we present a branch and bound framework that employs a column generation process to solve the proposed model. Our approaches, different from the literature, can either obtain an optimal VNE solution, or near-optimal solutions with guaranteed solution quality.

As VNE can be decomposed into two processes including node assignment (NA) and link mapping (LM), it is a common strategy to employ a divide and conquer design that sequentially addresses the NA and LM sub-problems. The non-optimality however trade-offs the simplicity of this design. We are hence motivated to design a decomposition process employs similar structure of NA and LM sub-problem while maintaining the optimality or near-optimality of the solution. Based on primal-dual analysis of the path-based ILP model, in this dissertation, we present an intelligent framework that allow feedback between NA and LM sub-problems, and evolvement towards the optimal solution or near-optimal solution with guaranteed performance.

Survivability is also critical to network virtualization, despite the extra complexity introduced to the VNE process. In this dissertation, we focus on the single facility node failure, and address the resulted Survivable Virtual Network Embedding problems from a network flow viewpoint. Specifically, we consider two variations of network flows: non-splitable flow and splitable flow, and propose both optimal and heuristic approaches for both cases. Our performance studies reveal interesting insights and the tradeoff between these two variations of flow mapping.

This work also presents a study on the motivation, architecture and model for optical-based network virtualization. The traffic explosion of the Internet calls for a high-speed automated optical substrate that can elastically allocate resources to support network services. In this dissertation, as the enabler, we present a four-layer architecture, and formally define and model the optical virtual network embedding problem.

7.2 Future Directions

Looking forward, our exploration in the area of network virtualization will be focusing on the following major topics.

First, the security in network virtualization. In a virtualized environment, data and/or computing are generally out-sourced to the third party and thus leading to inherent threats to the security goals. As a crucial component, however, security issues in network virtualization has only received limited attention [60] [61] [62] [63].

Second, as the power usage of the Internet has becoming an critical concern, it is also important to take the energy aspects into the consideration of virtual network embedding. Only few work has been devoted to this direction [64]. In our future work, we plan to explicitly assess the major power sources of network virtualization from various spectrums including, for example, application, facility resource type, node type (i.e., switch or facility node), resource quantity (e.g., bandwidth and computing resources) to present a framework that allows for energy-efficient virtual network embedding.

Third, in the area of optical network, the recent proposal on OFDM-based elastic optical networks is regarded as a promising solution to address the inefficiency of WDM networks in both energy and spectrum usage [29] [53] [54] [65] [66] [67] [68] [69]. OFDM-based elastic optical networks at the same time, brings extra complexity in the spectrum allocation including the spectrum-continuity and sub-carrier consecutiveness [54]. When these constraints are taken into account, the Optical Virtual Network Embedding problem is more challenging and deserves further study.

Fourth, in parallel with the efforts on the virtualization of the core of the Internet infrastructure, studies on the virtualization of various physical networks are also prominent. The counterpart study in wireless networks, for instance, are discussed in [70] where unique challenges of wireless network virtualization are elaborated. The topic of mobile cellular network virtualization has been addressed in [71], [72], [73]. In the area of sensor networks, the virtualization mainly focus on the virtualization of software platform to enable application sharing [20]. In our future study, we will also address main challenges associated with the virtualization of wireless and sensor networks as well as many other emerging networking paradigms (e.g., vehicular networks).

REFERENCES

[1] N. K. Chowdhury and R. Boutaba, “Network virtualization: State of the art and research challenges,” *IEEE Communications Magazine*, vol. 47, no. 7, pp. 20–26, 2009.

[2] T. Anderson, L. Peterson, S. Shenker, and J. Turner, “Overcoming the Internet impasse through virtualization,” in *Proceedings of the 3rd ACM Workshop on Hot Topics in Networks (HotNets-III)*, 2004, pp. 34–41.

[3] J. Turner and D. Taylor, “Diversifying the Internet,” in *Proceedings of IEEE GLOBECOM’05*, 2005, pp. 755–760.

[4] T. Anderson, L. Peterson, S. Shenker, and J. Turner, “Overcoming the Internet impasse through virtualization,” *IEEE Computer*, vol. 38, no. 4, pp. 34–41, April 2005.

[5] N. Feamster, L. Gao, and J. Rexford, “How to lease the Internet in your spare time,” *ACM SIGCOMM Computer Communication Review*, vol. 37, pp. 61–64, 2007.

[6] Xen, “http://http://www.xenproject.org/.”

[7] OpenFlow, “https://www.opennetworking.org//.”

[8] D. G. Andersen, “Theoretical approaches to node assignment,” 2002, unpublished Manuscript.

[9] N. K. Chowdhury, M. R. Rahman, and R. Boutaba, “Virtual network embedding with coordinated node and link mapping,” in *Proceedings of IEEE INFOCOM’09*, April 2009, pp. 783–791.

[10] J. Lischka and H. Karl, “A virtual network mapping algorithm based on subgraph isomorphism detection,” in *Proceedings of the 1st ACM workshop on Virtualized infrastructure systems and architectures*, 2009, pp. 81–88.

[11] Q. Hu, Y. Wang, and X. Cao, “Resolve the virtual network embedding problem: A column generation approach,” in *INFOCOM Mini-Conference*, 2013, pp. 1–5.

[12] ——, “Survivable network virtualization for single facility node failure: A network flow perspective,” *Optical Switching and Networking*, vol. 1, no. 1, pp. 1–22, 2013.

[13] M. R. Rahman, I. Aib, and R. Boutaba, “Survivable virtual network embedding,” in *Proceedings of NetWorking’10*, 2010, pp. 40–52.

[14] M. R. Rahman and R. Boutaba, “SVNE: Survivable virtual network embedding algorithms for network virtualization,” *IEEE Transactions on Network and Service Management*, December 2012.

[15] H. Yu, C. Qiao, V. Anand, X. Liu, H. Di, and G. Sun, “Survivable virtual infrastructure mapping in a federated computing and networking system under single regional failures,” in *Proceedings of IEEE GLOBECOM’10*, December 2010.

[16] H. Yu, V. Anand, C. Qiao, and H. Di, “Cost efficient design of survivable virtual infrastructure to recover from facility node failures,” in *Proceedings of IEEE ICC’11*, 2011.

[17] C. Qiao, B. Guo, S. Huang, J. Wang, T. Wang, and W. Gu, “A novel two-step approach to surviving facility failures,” in *Proceedings of IEEE/OSA OFC’11*, March 2011.

[18] R. Ramaswami and K. N. Sivarajan, *Optical Networks: A Practical Perspective*. Morgan Kaufmann Publishers, 2002.

[19] Y. Zhang, P. Chowdhury, M. Tornatore, and B. Mukherjee, “Energy efficiency in telecom optical networks,” *Communications Surveys Tutorials, IEEE*, vol. 12, no. 4, pp. 441–458, 2010.

[20] A. Wang, M. Iyer, R. Dutta, G. N. Rouskas, and I. Baldine, “Network virtualization: Technologies, perspectives, and frontiers,” *J. Lightwave Technol.*, vol. 31, no. 4, pp. 523–537, Feb 2013.

[21] "CPLEX optimizer:high-performance mathematical programming solver for linear programming, mixed integer programming, and quadratic programming," http://www.ilog.com/products/cplex/.

[22] "Gnu linear programming kit," http://www.gnu.org/software/glpk/.

[23] X. Cheng, S. Su, Z. Zhang, H. Wang, F. Yang, Y. Luo, and J. Wang, "Virtual network embedding through topology-aware node ranking," *ACM SIGCOMM Computer Communication Review*, vol. 41, pp. 38–47, April 2011.

[24] I. Fajjari, N. Aitsaadi, G. Pujolle, and H. Zimmermann, "Vne-ac: Virtual network embedding algorithm based on ant colony metaheuristic," in *IEEE International Conference on Communications*, June 2011, pp. 1–6.

[25] Z. Zhang, X. Cheng, S. Su, Y. Wang, K. Shuang, and Y. Luo, "A unified enhanced particle swarm optimization-based virtual network embedding algorithm," *International Journal of Communication Systems*, vol. 26, pp. 1054–1073, 2012.

[26] I. Houidi, W. Louati, D. Zeghlache, P. Papadimitriou, and L. Mathy, "Adaptive virtual network provisioning," in *Proceedings of the second ACM SIGCOMM workshop on Virtualized infrastructure systems and architectures (VISA)*, September 2010, pp. 41–48.

[27] S. Herker, A. Khan, and X. An, "Survey on survivable virtual network embedding problem and solutions," in *Proceedings of the Ninth International Conference on Networking and Services*, 2013, pp. 99–104.

[28] "Cisco visual networking index: Forecast and methodology, 2012-2017," http://www.cisco.com/.

[29] M. Jinno, Y. Tsukishima, H. Takara, B. Kozicki, Y. Sone, and T. Sakano, "Virtualized optical network (VON) for future Internet and applications," *IEEE Transactions of Communications*, vol. E93-B, no. 3, pp. 470–477, March 2010.

[30] R. Nejabati, E. Escalona, S. Peng, and D. Simeonidou, “Optical network virtualization,” in *Proceedings of ONDM’11*, Feb. 2011.

[31] R. Nejabati, S. Peng, and D. E. Simeonidou, “Role of optical network infrastructure virtualization in data center connectivity and cloud computing,” in *Optical Fiber Communication Conference/National Fiber Optic Engineers Conference*, pp.OM3E.1, 2013.

[32] Q. Zhang, W. Xie, Q. She, X. Wang, P. Palacharla, and M. Sekiya, “Rwa for network virtualization in optical wdm networks,” in *Optical Fiber Communication Conference/National Fiber Optic Engineers Conference 2013*, 2013, p. JTh2A.65.

[33] R. J. Vanderbei, *Linear Programming: Foundations and Extensions*. Kluwer Academic Publishers, 2001.

[34] M. Lubbecke and J. Desrosiers, “Selected topics in column generation,” *Operation Research*, vol. 53, pp. 1007–1023, 2005.

[35] B. Tiwana, M. Balakrishnan, M. K. Aguilera, H. Ballani, and Z. M. Mao, “Location, location, location! modeling data proximity in the cloud,” in *Proceedings of Hotnets’10*, October 2010.

[36] M. Grotschel, L. Lovasz, and A. Schrijver, “The ellipsoid method and its consequences in combinatorial optimization,” *Combinatorica*, vol. 4, no. 4, pp. 291–295, 1984.

[37] M. E. M. Saad and Z.-Q. Luo, “A lagrangean decomposition approach for the routing and wavelength assignment in multifiber wdm networks,” in *Proceedings of IEEE GLOBECOM’02*, vol. 3, 2002, pp. 2818–2822.

[38] Y. Zhu and M. Ammar, “Algorithms for assigning substrate network resources to virtual network components,” in *Proceedings of IEEE INFOCOM’06*, April 2006, pp. 1–12.

[39] M. Yu, Y. Yi, J. Rexford, and M. Chiang, “Rethinking virtual network embedding: Substrate support for path splitting and migration,” *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 17–29, April 2008.

[40] N. F. Butt, M. Chowdhury, and R. Boutaba, "Topology-awareness and reoptimization mechanism for virtual network embedding," in *Proceedings of the 9th IFIP NETWORKING Conference*, May 2010, pp. 27–39.

[41] S. Qing, Q. Qi, J. Wang, T. Xu, and J. Liao, "Topology-aware virtual network embedding through bayesian network analysis," in *IEEE GLOBECOM*, 2012, pp. 2621–2627.

[42] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows: Theory, Algorithms, and Applications*. Englewood Cliffs, NJ: Prentice Hall, 1993.

[43] Q. Hu, Y. Wang, and X. Cao, "Resolve the virtual network embedding problem: A column generation approach," in *IEEE INFOCOM*, 2013, pp. 410–414.

[44] F. S. Hillier and G. J. Lieberman, *Introduction to Operations Research*. McGraw-Hill, 2010.

[45] J. Hooker, "Logic-based benders decomposition," *Mathematical Programming*, vol. 96, p. 2003, 1995.

[46] R. Martin, *Large Scale Linear and Integer Optimization: A unified appproach*. Boston: Kluwer Academic, 1999.

[47] Y. Dinitz, N. Garg, and M. X. Goemans, "On the single-source unsplittable flow problem," in *Proceedings of the 39th Annual Symposium on Foudations of Computer Science*, 1998, pp. 290–299.

[48] G. Baier, E. Köhler, and M. Skutella, "On the k-splittable flow problem," in *Proceedings of the 10th Annual European Symposium on Algorithms*, ser. ESA '02. London, UK: Springer-Verlag, 2002, pp. 101–113. [Online]. Available: http://dl.acm.org/citation.cfm?id=647912.740662

[49] Q. Hu, Y. Wang, and X. Cao, "Resolve the virtual network embedding problem: a column generation approach," in *Proceedings of IEEE INFOCOM Mini-Conference' 09*, 2013, pp. 1–5.

[50] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, “Openflow: enabling innovation in campus networks,” *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, Mar 2008.

[51] C. Qiao and M. Yoo, “Optical burst switching (obs) - a new paradigm for an optical Internet,” *J. High Speed Networks*, vol. 8, no. 1, pp. 69–84, 1999.

[52] I. Chlamtac, A. Ganz, and G. Karmi, “Lightpath communications: an approach to high bandwidth optical WAN’s,” *IEEE Transcations on Communications*, vol. 40, no. 7, pp. 1171–1182, 1992.

[53] M. Jinno, H. Takara, B. Kozichi, Y. Tsukishima, and Y. Sone, “Spectrum-efficient and scalable elastic optical path network: Architecture, benefits, and enabling technologies,” *IEEE Communications Magazine*, vol. 47, pp. 66–73, 2009.

[54] Y. Wang, X. Cao, and Y. Pan, “A study of the routing and spectrum allocation in spectrum-sliced elastic optical path networks,” in *INFOCOM, 2011 Proceedings IEEE*, 2011, pp. 1503–1511.

[55] L. Sahasrabuddhe and B. Mukherjee, “Light trees: optical multicasting for improved performance in wavelength routed networks,” *Communications Magazine, IEEE*, vol. 37, no. 2, pp. 67–73, 1999.

[56] J. P. A. Gadkar and V. Vokkarane, “Multicast overlay for highbandwidth applications over optical wdm networks,” *IEEE/OSA Journal of Optical Communications and Networking*, vol. 4, no. 8, 2012.

[57] M. Ali and J. Deogun, “Power-efficient design of multicast wavelengthrouted networks,” *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 10, 2000.

[58] S. Chaudhuri, G. Hjilmtjsson, and J. Yates, “Control of lightpaths in an optical network,” Jan. 2000, optical Internetworking Forum OIF2000.04, IETF Internet Draft.

[59] R. Asthana, Y. Singh, and W. Grover, "p-cycles: An overview," *Communications Surveys Tutorials, IEEE*, vol. 12, no. 1, pp. 97–111, 2010.

[60] L. R. Bays, R. R. Oliveira, L. S. Buriol, M. P. Barcellos, and L. P. Gaspary, "Security-aware optimal resource allocation for virtual network embedding," in *2012 Proceedings of the 8th International Conference on Network and Service Management*, pp. 378–384.

[61] C. Xing, J. Lan, and Y. Hu, "Virtual network with security guarantee embedding algorithms," *Journal of Computers*, vol. 8, no. 11, pp. 2782–2787, November 2013.

[62] S. Liu, Z. Cai, H. Xu, and M. Xu, "Security-aware virtual network embedding," in *2014 IEEE International Conference on Communications*.

[63] P. Chau and Y. Wang, "Security-awareness in network virtualization: A classified overview," in *2014 IEEE MASS*, pp. 545–550.

[64] J. Botero, X. Hesselbach, M. Duelli, D. Schlosser, A. Fischer, and H. de Meer, "Energy efficient virtual network embedding," *Communications Letters, IEEE*, vol. 16, no. 5, pp. 756–759, May 2012.

[65] M. Jinno, B. Kozichi, H. Takara, A. Watanabe, Y. Sone, T. Tanaka, and A. Hirano, "Distance-adaptive spectrum resource allocation in spectrum-sliced elastic optical path network," *IEEE Communications Magazine*, vol. 48, pp. 138–145, 2010.

[66] Y. Sone, A. Watanabe, W. Imajuku, Y. Tsukishima, B. Kozicki, H. Takara, and M. Jinno, "Highly survivable restoration scheme employing optical bandwidth squeezing in spectrum-sliced elastic optical path SLICE network," in *Proceedings of OFC'09*, pp. 1–3.

[67] K. Christodoulopoulos, I. Tomkos, and E. Varvarigos, "Routing and spectrum allocation in OFDM-based optical networks with elastic bandwidth allocation," in *Proceedings of IEEE GLOBECOM*, 2010, pp. 1–6.

[68] M. Klinkowski and K. Walkowiak, “Routing and spectrum assignment in spectrum sliced elastic optical path network,” *IEEE Communications Letters*, vol. 15, no. 8, pp. 884–886, Aug. 2011.

[69] M. Jinno, H. Takara, and B. Kozichi, “Dynamic optical mesh networks: drivers, challenges, and solutions for the future,” in *Proceedings of ECOC’09*, pp. 1–5.

[70] S. Paul and S. Seshan, “Technical document on wireless virtualization,” Sep. 2006, tech. Rep., GDD-06-17.

[71] M. Hoffmann and M. Staufer, “Network virtualization for future mobile networks: General architecture and applications,” in *Communications Workshops (ICC), 2011 IEEE International Conference on*, June 2011, pp. 1–5.

[72] R. Kokku, R. Mahindra, H. Zhang, and S. Rangarajan, “Nvs: a virtualization substrate for wimax networks,” in *Proceedings of the sixteenth annual international conference on Mobile computing and networking*, Sep. 2010, pp. 20–24.

[73] Y. Zaki, L. Zhao, C. Görg, and A. Timm-Giel, “Lte mobile network virtualization,” *Mobile Networks and Applications*, vol. 16, no. 4, pp. 424–432, Aug. 2011.