Virtual Network Embedding with Path-based Latency Guarantees in Elastic Optical Networks

by

Sepehr Taeb

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2019

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

Most parts of this thesis appeared in this publication:

S. Taeb, N. Shahriar, S.R. Chowdhury, M. Tornatore, R. Boutaba, J. Mitra, and M. Hemmati. Virtual Network Embedding with Path-based Latency Guarantees in Elastic Optical Networks. IEEE International Conference on Network Protocols (ICNP). Chicago, IL, USA, October 7-10, 2019.

Chapter 5 borrows content from [45].

Abstract

Elastic Optical Network (EON) virtualization has recently emerged as an enabling technology for 5G network slicing. A fundamental problem in EON slicing (known as Virtual Network Embedding (VNE)) is how to efficiently map a virtual network (VN) on a substrate EON characterized by elastic transponders and flexible grid. Since a number of 5G services will have strict latency requirements, the VNE problem in EONs must be solved while guaranteeing latency targets. In existing literature, latency has always been modeled as a constraint applied on the virtual links of the VN. In contrast, we argue in favor of an alternate modeling that constrains the *latency of virtual paths*. Constraining latency over virtual paths (vs. over virtual links) poses additional modeling and algorithmic challenges to the VNE problem, but allows us to capture end-to-end service requirements. In this thesis, we first model latency in an EON by identifying the different factors that contribute to it. We formulate the VNE problem with latency guarantees as an Integer Linear Program (ILP) and propose a heuristic solution that can scale to large problem instances. We evaluated our proposed solutions using real network topologies and realistic transmission configurations under different scenarios and observed that, for a given VN request, latency constraints can be guaranteed by accepting a modest increase in network resource utilization. Latency constraints instead showed a higher impact on VN blocking ratio in dynamic scenarios.

Acknowledgements

I would like to thank my supervisor, Raouf Boutaba, for his persistent support intellectually, emotionally, and financially. This thesis would not be possible without his guidance.

I extend my thanks to the my thesis committee members, Omid Abari and Samer Al-Kiswany, for their review, and feedback.

I would also like to express my special gratitude to Nashid Shahriar and Shihabur Rahman Chowdhury for their constant help over the course of this project. I would like to extend my appreciation to Massimo Tornatore who provided his expertise and invaluable feedback on this work.

Above all, I owe my deepest gratitude to my parents, Parisa and Hossein, for their unconditional love and support along this way.

This work benefitted from the use of the CrySP RIPPLE Facility at the University of Waterloo.

Table of Contents

List of Tables

List of Figures

List of Acronyms

BER Bit Error Rate 14

BV-OXC Bandwidth-Variable Optical Cross-Connect 14, 16

BVT Bandwidth-Variable Transponders 13

CD Chromatic Dispersion 15

DCF Dispersion Compensating Fiber 14, 15

DCM Dispersion Compensation Module 15

DWDM Dense Wavelength Division Multiplexed 2, 5, 6

EDFA Erbium-Doped Fiber Amplifier 15

EON Elastic Optical Network 2–11, 13–16, 20, 22–24, 26, 35, 36, 38, 40, 42

FBG Fiber Bragg Gratings 15

FEC Forward Error Correction 3, 6, 8–15, 42

ILP Integer Linear Program 4, 18, 32, 36–38, 42

LNR Link to Node Ratio 33, 36, 38–41

OTN Optical Transport Network 2, 12–14, 16

ROADM Reconfigurable Optical Add-Drop Multiplexer 13, 14, 16

SLink Substrate Link 7, 11, 16, 17, 19, 23, 27, 30, 32, 37, 40–42

SN Substrate Network 1, 3, 17, 32, 35, 37, 40, 41

SNode Substrate node 7, 11, 12, 17, 32, 43

SPath Substrate Path 7, 12, 16–19, 23, 24, 26, 27, 30, 31, 34–42

VCAT Virtual Concatenation 2, 12, 16

VLink Virtual Link 7, 12, 16–24, 26, 27, 32, 34–42

VN Virtual Network 1–4, 8–10, 12, 13, 17, 20–24, 32–37, 39–43

VNE Virtual Network Embedding 1–5, 8–10, 22, 43

VNode Virtual Node 7, 12, 17, 22, 32–34, 43

VPath Virtual Path 12, 22, 24, 34

Chapter 1

Introduction

Ultra-low latency communication is an important quality-of-service (QoS) requirement for many emerging applications such as intelligent transportation, industry automation, immersive media experience through virtual and augmented reality, online multi-player gaming, and tactile Internet [26,31,38,53]. Additionally, safety, real-time control, and healthcare related applications can have life-threatening consequences if their stringent latency requirements are not met [38]. Latency also significantly impacts the revenue generated by services such as electronic commerce and high-frequency trading [3,41,48]. Consequently, latency-sensitive applications have become one of the major business drivers for the development of the fifth-generation (5G) mobile networks [5, 25, 31, 35]. A key enabling technology for deploying latency-sensitive applications in 5G networks is network virtualization (*aka* network slicing) [19], which allows the instantiation of one or more virtual networks (VNs) with dedicated substrate resources to guarantee latency between application end nodes.

A key challenge in instantiating VNs for latency-sensitive applications is to devise a mechanism to efficiently map VN nodes and links on the substrate network (SN). This problem, known as Virtual Network Embedding (VNE) [16], has been extensively studied in the past, with a particular focus on ensuring bandwidth and reliability requirements [18]. Less attention has been devoted to the VNE problem with latency constraints. In existing literature, latency requirements have been modeled as constraints applied to the virtual links of the VN (*i.e.*, each virtual link is mapped to satisfy a given latency target) [6, 8, 23, 27, 46]. However, we argue in the favor of an alternate latency constraint model, namely the virtual path based latency constraint, where the VN request comes with a VN topology and a set of virtual paths with end-to-end latency requirements. These virtual paths can come from the service provider's own routing and/or fault tolerant policy while taking the application semantics into context. Applications typically require that a given latency target is achieved along an entire path between application end nodes. For instance,

two virtual nodes in a VN can represent two trading sites running an online trading service. In this case, the network operator must compute one or more virtual paths between these two sites while guaranteeing stringent latency requirements of the service.

Given that virtual paths are merely a sequence of virtual links, one can argue that latency constraints on the given virtual paths can be modeled as a system of linear equations consisting of latency constraints on the virtual links. However, there is no guarantee that the given constraints will not form an under-determined system. In that case, there can be many ways of allocating virtual path latencies to virtual link latencies. Finding the optimal allocation that ensures minimum cost VN embedding further increases the complexity of the already computationally complex problem of performing VN embedding on an Elastic Optical Network (EON) [45]. Therefore, we propose a mathematical model that performs VN embedding in a way such that it guarantees optimality while explicitly computing per virtual link latencies. Note that we allow to split virtual link demands on multiple EON paths (similar to [37]) while solving the VNE over EON problem with latency guarantee on the virtual paths. This is particularly useful when no substrate path can yield a data-rate large enough to accommodate a virtual link's demand. Such multi-path provisioning can be supported by Virtual Concatenation (VCAT) in Optical Transport Network (OTN) [7] or the bonding capabilities of FlexE [1].

From a resource allocation perspective, per-virtual-path latency constraints have the advantage of granting more flexibility in the selection of substrate paths for embedding compared to per-virtual-link latency constraints. For instance, in Fig. 1.1, the latency bound on virtual link *qr* can be varied as long as the latency constraints on virtual paths *p-r-q* and *q-r-s* are not violated. Hence, more options to embed virtual link qr become available if we assume per-path latency constraints, compared to having a fixed latency constraint on *qr*. However, this flexibility comes with the challenge of cleverly distributing the latency budgets of virtual paths to constituent virtual links in a way that results in the optimal selection of substrate paths for embedding. To the best of our knowledge, no previous work has addressed the VNE problem with latency constraints on virtual paths.

The networking infrastructure supporting 5G applications can span different technology domains such as wireless radio, access/core/metro transport networks, and multi-tier data centers [19]. Each technology domain exhibits its own latency characteristics. In this thesis, we focus on transport networks, which constitute the largest segments in a telecommunication network infrastructure. Traditional dense wavelength division multiplexed (DWDM) optical networks (also called "fixed grid" optical networks), have been for a long time a dominant technology in the transport segment thanks to their high-bandwidth and low-latency characteristics. Recently, network operators are adopting EONs to overcome the limitations of DWDM networks, such as inflexible and coarse-grained resource allocation [10, 20]. EONs have the capability to allocate an arbitrary number of spectrum slices for right-size spectrum allocation to customer needs

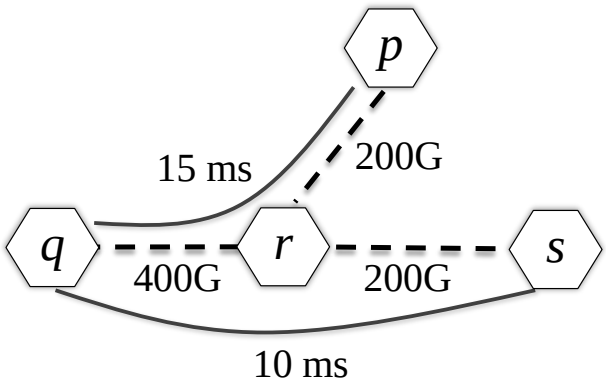


Figure 1.1: VN with latency requirement on virtual paths (marked by solid lines)

(i.e., EONs employ a flexible grid, as opposed to the traditional fixed grid, to allocate optical spectrum slices). They also leverage the advances in coherent-transmission technology allowing the tuning of transmission parameters such as baud rate, modulation format and forward error correction (FEC) overhead. In what follows, we will consider EON as the underlying SN for our problem.

EON virtualization has gained interest from the research community in recent times, especially because of its importance in 5G network slicing [10]. A fundamental problem in EON virtualization is to efficiently map a VN on the substrate EON [18]. VNE has been extensively studied for layer-2/3 electrical and fixed-grid optical networks [18]. However, VNE over EON introduces unique challenges and complexities due to the large number of flexible transmission parameters made available by EONs [45]. Several proposals for optimal resource allocation in EONs have considered tuning all or a subset of these transmission parameters for satisfying bandwidth requirements [22, 32, 45]. Even some research effort has been made towards addressing latency guaranteed VNE for layer-2/3 networks (a detailed literature review is presented in Chapter 2). However, to the best of our knowledge no work in the EON virtualization literature has considered satisfying path-based latency requirements.

1.1 Contributions

The main research contributions of this thesis are as follows:

- We address the problem of VNE over EON with latency guarantees on a set of given virtual paths, while considering all the configurable transmission parameters on EONs, and

allowing virtual link demands to be split on multiple substrate EON paths. We propose a latency model in an EON by investigating the latency contributors both at EON node and lightpath. To do so, we consider the components at EON nodes and along the lightpaths, and we discuss the latency incurred by each of these components to transport data through an EON.

- We present an Integer Linear Program (ILP) formulation for the VNE problem with path-based latency requirements over EON, which enables us to solve this problem optimally. In addition, due to the intractability of the ILP solution, we propose a heuristic algorithm to tackle larger instances of this problem.

- We perform extensive evaluations of our proposed solutions using real network topologies from [2] under different evaluation scenarios. Our simulation results show that the heuristic incurs only 2.5% and 0.8% additional cost on average compared to the optimal solution for fixed- and flex-grid EONs, respectively, while executing three to four orders of magnitude faster. Moreover, we observed that guaranteeing latency constraints of a VN causes a modest increase in resource utilization, while it results in a more significant impact on VN blocking ratio in dynamic scenarios.

1.2 Thesis Organization

The rest of this thesis is organized as follows. Chapter 2 presents the necessary background for this thesis including an overview of the EON architecture, and resource allocation over EONs. The chapter also provides a discussion of the related works. In Chapter 3, we present the problem definition and propose a latency model for EONs. We present an ILP formulation for the problem in Chapter 4. We propose a heuristic algorithm in Chapter 5. In Chapter 6, we evaluate the ILP and heuristic implementation for different scenarios. Finally, we conclude this thesis by summarizing the main thesis contributions, and outlining some possible future work in Chapter 7.

Chapter 2

Background

In this chapter, we discuss the evolution of the optical network architecture from DWDM to EON in Section 2.1.1. A discussion then follows about the transmission configurations and resource allocation for VNE over EONs in Section 2.1.2 and Section 2.1.3, respectively. In Section 2.2, we present the related works under three categories: 1) works that address the VNE over EON (in Section 2.2.1); 2) works that study latency-guaranteed VNE (in Section 2.2.2), and 3) works that discuss the latency aspects in optical networks (in Section 2.2.3).

2.1 Optical Networks

2.1.1 Evolution from DWDM to EON

Optical networks have been the supporting technology for deploying transport networks for a long time due to their high-bandwidth and low-latency communication capabilities. Fixed-grid dense wavelength division multiplexing (DWDM) has been the de facto technology used for optical networking. However, DWDM is falling short of meeting today's capacity requirements for multiple reasons. First, coarse-grained spectrum allocation in DWDM leads to wasted capacity, *e.g.*, fitting a 40Gbps Ethernet signal into a 50Ghz size grid wastes 60% of slice capacity. Second, fixed-grid DWDM is not flexible enough to meet the growing need for supporting data-rates beyond 100Gbps [20]. In addition, DWDM devices cannot dynamically adjust transmission properties (*e.g.*, , spectrum of wavelength, bit rate, modulation scheme) according to the rapid scaling need of a 5G network slice. These shortcomings of fixed-grid DWDM have led to the evolution towards flex-grid optical networks. The more recent flex-grid approach divides the

frequency spectrum into smaller (*e.g.*, 12.5Ghz or 6.25Ghz) slices and allows allocation of an arbitrary number of slices to right size spectrum allocation to customer needs. Flex-grid, combined with the recent advances in coherent-transmission technology, can bring unprecedented elasticity in tuning transmission parameters such as baud rate, modulation format and forward error correction (FEC) overhead. In addition, emerging transponder technologies such as bandwidth-variable transponders and multi-flow transponders offer unprecedented capability of transmitting and receiving a traffic demand into multiple elastic optical flows, operating at independent bit rates, that can be routed independently [37]. This new generation of optical networks is usually referred to as Elastic Optical Networks (EONs) [10, 20]. EON capable devices also offer the proper abstraction to virtualize an optical network into a number of network slices as demanded by the 5G architecture. Consequently, a higher number of network slices can be instantiated compared to DWDM, while taking the QoS requirement of the network slices into account.

2.1.2 Transmission Configurations in EON

There are three main parameters that specify a transmission configuration in EONs: 1) baud rate (symbol rate) which defines the number of symbols generated by a transponder at the source node, 2) modulation format which specifies the number of bits transported by each symbol. 3) forward error correction (FEC) which can be used at the destination node to detect and correct the errors in the transmitted data. The amount of FEC overhead is specified by its percentage compared to the total transmitting data.

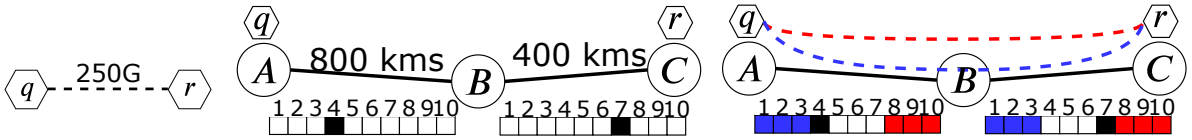
Some possible transmission configurations are presented in Table 2.1. Each row represents a configuration along with its transmitting data rate, its reach, and the number of slices that is needed for the configuration assuming that width of each spectrum slice is 12.5 GHz. For example, the first configuration generates 56.5 billion symbols per second and each of them transmits two bits (QPSK modulation format), and 33% of the data are used for FEC. Each configuration has some limitation in terms of the physical distance that a signal generated with a specific transmission configuration could travel without degradation of the signal quality which is known as the reach of that configuration. It is worth mentioning that the reach is dependent on both FEC overhead and modulation format. By increasing the FEC overhead the reach of a configuration also increases [21] whereas the actual data rate decreases. For example, in Table 2.1, the second and third configurations are the same in terms of modulation format and data rate; however, the second configuration has a higher FEC overhead. Consequently, the second configuration has higher reach. By contrast, going to a higher modulation format (which transmits more bits per symbol) increases the data rate, and reduces the reach of a configuration [12]. E.g., in Table 2.1, the fourth configuration has a higher modulation format compared to the first one, and it has higher data rate and lower reach.

Table 2.1: Reach Table Example

	Data Rate (Gb/s)	Modulation Format	FEC	Baud Rate (Gbaud/s)	Operating Bandwidth (GHz)	No. of Slices	Reach (Km)
1	150	QPSK	33%	56.5	62.5	5	1800
2		8QAM	33%	40	50	4	1400
3		8QAM	20%	32	37.5	3	1200
4	250	8QAM	33%	64	75	6	1400
5		16QAM	33%	48	50	4	1000

2.1.3 Resource Allocation for VNE over EON

Embedding a virtual link (VLink) in an EON requires choosing one or more substrate path(s) (SPath(s)) (considering splitting over multiple SPaths); transmission configuration satisfying the reach, considering the SPath length; and spectrum slice allocation along each SPath satisfying the contiguity and continuity constraints. That is, each split which uses a specific transmission configuration should use a consecutive set of spectrum slices on each substrate link (SLink) along a lightpath (contiguity) and the set of assigned spectrum slices should be the same along a lightpath (continuity). Fig. 2.1 shows the embedding of a VLink request qr (Fig. 2.1(a)) over substrate EON in Fig. 2.1(b). The virtual nodes (VNodes) q and r are mapped to substrate nodes (SNodes) A and C, respectively. Each SLink has 10 spectrum slices with 12.5 GHz slice width. Before embedding VLink qr the 4th and 7th spectrum slices are occupied on SLinks AB and BC, respectively. To embed the VLink qr with 250 Gbps using the transmission configurations in Table 2.1, one can either use one split with 250 Gbps data rate or two splits with 150 Gbps each. Since the length of the SPath *A-B-C* is 1200 kms the 5th configuration, which has a 1000 km reach, could not be used to provision the demand. Moreover, there is no set of 6 consecutive spectrum slices which are unoccupied on both AB and BC; hence, provisioning this demand using 4th transmission configuration is infeasible. However, the demand could be satisfied using two splits of 150 Gbps each using the 3rd transmission configuration (Fig. 2.1(c)). Although in Fig. 2.1(c), VLink qr demand is split over the same SPath, splitting over distinct SPaths is also possible.



(a) A VLink request (b) State of the SN before embedding VLink qr (c) State of the SN after embedding VLink qr using two splits

Figure 2.1: VLink embedding in EON

2.2 Related Works

2.2.1 VNE over EON

Several variants of the VNE problem have been extensively studied over the last decade for layer 2/3 network virtualization [18]. The problem has been investigated in the context of fixed-grid optical networks [12] and, more recently, in the context of EONs [22, 32, 54]. Two variants of the VNE problem over EON, namely, transparent VNE and opaque VNE have been investigated in [22] and [32], respectively. In the transparent case, all the virtual links of a VN must use the same spectrum for their substrate links, while in the opaque VNE case only the substrate links in the embedding of the same virtual link must use the same spectrum. While refs. [22] and [32] proposed a formulation based on multi-commodity flow, Ref. [54] uses pre-computed path based formulations for the VNE over EON problem. However, none of these works leverages the full flexibility of an EON, namely combining different modulation formats, FECs, and baud rates to allow multiple transmission reaches. However, none of these works leverages the full flexibility of an EON, namely combining different modulation formats, FECs, and baud rates to allow multiple transmission reaches. Several works have considered the impact of these transmission parameters in isolation, however, not in the context of the VNE problem (see e.g., [30] for distance-adaptive modulation formats, [42] for variable baud rates, and [43, 44] for variable FEC redundancy). More recently, the work presented in [45] investigated the flexibility offered by EON for different transmission parameters can be leveraged for solving the VNE problem.

2.2.2 Latency-guaranteed VNE on Layer-2/3 Networks

A number of research efforts have been dedicated to investigate the VNE problem on layer-2/3 substrate networks while considering various forms of delay constraints such as per-virtual link delay [6,8,23,27,46] and differential delay in a multi-cast virtual network [6,52]. Chochlidakis *et*

al., , addresses the delay-constrained VNE problem in the context of mobile networks, where the VN request has a tree structure representing communication between edge routers and a single gateway node [14]. While computing the VN embedding, [14] also computes provisions paths between edge routers so that users can seamlessly hand-off between them. Unlike other works in the area which consider delay as a single quantitiy, [14] models substrate network delay as a combination of propagation delay and queuing delay. Queuing delay, which was modeled as a hyperbolic function of residual link capacity in the authors' earlier work [15], is then approximated using a piecewise linear function for simplifying the optimization model. A number of different objectives have been considered across these works, including, minimizing resource (*i.e.*, substrate node and/or link resource) provisioning cost [6, 14, 27, 46, 52], maximizing acceptance ratio [8, 52], balancing delay across virtual links [28], and minimizing energy consumption [23]. Different techniques have been leveraged to solve different variations of the delay-constrained VNE problem, including, integer programming [6, 27, 28], markov reward based node similarity ranking [8], meta-heuristic algorithm [6], network segmentation [23, 46], projected sub-gradient method [14]. In addition to solving delay-constrained VNE problem, Inführ *et al.*, , also proposes a benchmarking mechanism to generate problem instances for such problems. They propose generating four classes of VNE requests representing different applications, namely web applications, streaming, P2P file transfer, and VoIP application, by setting different ranges of values for bandwidth and delay constraints. However, most works in delay-constrained VNE abstract all components of substrate network delay into one metric, *i.e.*, physical link delay. An elaborate discussion on the contributing factors to layer-2/3 substrate network delay and literature survey on how they have been modeled in the past is presented in [34].

2.2.3 Latency Considerations in Optical Network

Several research papers and industry white papers have discussed different sources of latency in an optical network [9, 26, 29, 36, 40]. According to the literature latency in optical network stem from various technology domains such as physical layer operations [29], components inside an optical switching element [9], transport layer processing [26, 36, 40]. This body of work will be helpful in determining the latency components that should be considered while modeling the delay-constrained VNE problem for EON. Furthermore, the numeric ranges reported for different latency components will help us in deciding if certain latency components can be omitted to keep the model complexity at a reasonable level.

Tanimura *et al.*, proposed the architecture of a programmable optical transceiver that enables simultaneous optimization of multiple tunable transmission programmable parameters, namely modulation format, symbol rate, power allocation, and FEC overhead for satisfying throughput, signal quality, and latency requirements [47]. They argue that in addition to propagation delay,

other transmission parameters (*e.g.*, FEC overhead computation) overhead computation can have impact on latency of an end-to-end optical path. An analytical model to derive a transmission configuration that satisfies a given throughput and latency requirement is also presented in [47].

To the best of our knowledge, delay-constrained VNE for EON substrate with flexible transmission parameters has not been addressed in the literature. Furthermore, the literature on delay-constraint VNE over layer-2/3 networks consider per virtual link delay constraint. In contrast, we propose a novel delay modeling for the VN, *i.e.*, virtual path based delay constraints, which is more suitable to represent the requirement of emerging ultra-low latency applications.

Chapter 3

Mathematical Model and Problem Statement

In this chapter, we first present a mathematical model of the inputs and latency characterization of different EON components followed by a formal statement of the problem.

3.1 Substrate EON

The substrate EON (SN) is an undirected graph $G = (V, E)$, where V and E are the set of substrate optical nodes (SNodes) and substrate optical links (SLinks), respectively. Without loss of generality, we assume the optical nodes to be colorless, directionless, and contentionless [4]. We also assume the SLinks to be bi-directional, *i.e.*, adjacent optical nodes are connected by one optical fiber in each direction. The optical frequency spectrum on each SLink $e = (u, v) \in E$ is divided into equal-width slices represented by the set S and enumerated as $1, 2 \ldots |S|$. In what follows, we use the term frequency/spectrum slot and slice interchangeably. $\mathcal{P}$ and $\mathcal{P}_{uv}^{k} \subset \mathcal{P}$ represents the set of all paths in G and the set of k-shortest paths between nodes $u, v \in V$, respectively. The two endpoints of a path $p \in \mathcal{P}$ are represented by $src(p)$ and $dst(p)$. The number of SLinks and the physical length of a path p in kilometers are represented by $|p|$ and $len(p)$, respectively. We use the binary variable δ_{pe} to denote the association between a path $p \in \mathcal{P}$ and any link $e \in E$.

The following transmission parameters can be configured on a path p with physical length $len(p)$ to enable data transmission with different data-rates $d \in \mathcal{D}$: *baud-rate* or *symbol-rate*, b, *modulation format*, m, and *forward error correction code* (FEC) *overhead*, f, selected from the

set of possible values $\mathcal{B}$, $\mathcal{M}$, and $\mathcal{F}$, respectively. We use a tuple $t = (d, b, m, f) \in \mathcal{T} = (\mathcal{D} \times \mathcal{B} \times \mathcal{M} \times \mathcal{F})$ to represent a transmission configuration that dictates the combination of $b \in \mathcal{B}$, $m \in \mathcal{M}$, and $f \in \mathcal{F}$ that can be used to yield a data-rate $d \in \mathcal{D}$. For the sake of representation we use $t^{(d)}, t^{(b)}, t^{(m)}$, and $t^{(f)}$ to denote the data-rate, baud-rate, modulation format, and FEC overhead of a configuration $t \in \mathcal{T}$. A *reach table* $\mathcal{R}$, computed based on physical layer characteristics, specifies the maximum length of a path (*i.e.*, the *reach* r_t) capable of retaining a satisfactory optical signal to noise ratio when configured according to a transmission configuration $t \in \mathcal{T}$. Finally, n_t denotes the number of slices required to accommodate a transmission configuration $t \in \mathcal{T}$, which is dependent on the parameters of t.

3.2 Virtual Network Request

The virtual network requests are made in the form of an undirected graph $\bar{G} = (\bar{V}, \bar{E})$, where $\bar{V}$ and $\bar{E}$ are the set of virtual nodes (VNodes) and virtual links (VLinks), respectively. The function $\tau : \bar{V} \rightarrow V$ represents VNode to SNode mapping and is an input to our problem (a common assumption for optical network virtualization [51]). Each virtual link $\bar{e} \in \bar{E}$ has a bandwidth requirement $\bar{\beta}_{\bar{e}}$. We allow the VLinks to be mapped to multiple substrate paths (SPaths) (similar to [37, 45]), each with a lower data-rate than $\bar{\beta}_{\bar{e}}$, since the reach becomes smaller for higher data-rates (*e.g.*, more than 400Gbps) limiting the number of usable paths. However, we limit the number of VLink splits to maximum q (≥ 1). Such multi-path provisioning is supported by technologies such as Virtual Concatenation (VCAT) in Optical Transport Network (OTN) [7] or bonding capabilities of FlexE [1].

3.3 Latency Constraints

We assume there exists a function $latency(.)$, which, when applied on any path, either virtual or substrate, returns the latency of that path. Let, $\bar{a}$ be a loop-free path in the VN (*i.e.*, a VPath). A VPath can have one (*i.e.*, representing a single VLink) or more VLink(s), therefore, $|\bar{a}| >= 1$. We represent the latency budget of a VPath using a tuple $\ell = (\bar{a}, L) \in (P_{\bar{G}} \times \mathbb{R}^+)$, where $P_{\bar{G}}$ is the set of all paths in $\bar{G}$ that require a bounded latency as mandated by the service provider and $\mathbb{R}$ represents the set of all positive real numbers. The tuple ℓ implies that after the embedding the latency of the VPath $\ell^{(\bar{a})}$ should not exceed $\ell^{(L)}$, *i.e.*, $latency(\ell^{(\bar{a})}) \leq \ell^{(L)}$, where $latency(\ell^{(\bar{a})}) = \sum_{\bar{e} \in \bar{a}} latency(\bar{e})$ and $latency(\bar{e})$ is the latency of the VLink $\bar{e}$. All the latency constraints of a VN request are represented by the set of such tuples $\mathcal{L}(\bar{G})$.

3.4 Latency Model

We present a latency model that captures different key latency contributors in an EON. The latency contributors can be broadly classified into those at EON nodes and those on a lightpath (Table 3.1). This latency model will be used to compute the latency perceived along a lightpath, which in turn will help determine the usable lightpaths for the VN embedding. In the following, we discuss the delay components under each category in detail.

Table 3.1: Latency contributors in EON

Node latency $2(L_{txp} + L_{fec})$	
OTN/FlexE elements (L_{otn})	Negligible
Transponders (L_{txp})	30 ns
FEC processing (L_{fec})	10 μs (std.), 150 μs (super)
Path latency = $len(p)L_{prop} + n_{amp}L_{amp} + (\|p\| + 1)L_{roadm}$	
Fiber propagation (L_{prop})	4.9 μs/km
Re-generators (L_{rgn})	Not considered
Amplifiers (L_{amp})	150 ns/unit
CD compensation (L_{dcf})	Not considered
ROADMs & BV-OXCs (L_{roadm})	O(nanoseconds)

3.4.1 Latency contributors at an EON node

We assume a multi-layer flexible node architecture as presented in [33], that consists of a set of OTN/FlexE line-cards (with or without a FEC module), a set of bandwidth-variable transponders (BVTs), and a reconfigurable optical add-drop multiplexer (ROADM). A transparent lightpath goes through all these components only at the source and destination EON nodes of the lightpath. At intermediate nodes, also known as the bypass nodes, the lightpath only passes through ROADMs bypassing OTN/FlexE elements and BVTs. Therefore, we will consider the latency contributed by ROADMs as a latency component at lightpath level to be discussed in Section 3.4.2.

OTN/FlexE elements One potential source of latency is due to processing and/or congestion at OTN/FlexE nodes for virtual link demand splitting/merging. In the case of FlexE, the FlexE Shim maps the FlexE Client(s) to a group of bonded Ethernet signals, each of which is carried

independently over the EON. In the destination node, another FlexE Shim is responsible for creating the FlexE Client(s) out of the bonded Ethernet signals [1]. In OTN, a mapper/demapper converts higher layer signal(s) to OTN frame(s) and vice versa. Research literature on the subject reports that the processing delay at FlexE shim or OTN mapper/demapper is negligible compared to optical transponder processing delay, FEC processing delay, and propagation delay along the fibers [36, 40]. Furthermore, OTN is a deterministic transport technology, therefore, there is no queuing delay due to congestion at the nodes. Similarly, FlexE nodes can be fully scheduled with static resource allocation, thereby avoiding any congestion related delay altogether [40]. Therefore, we ignore the latency incurred by OTN/FlexE nodes in our model.

Transponders (BVTs in case of EONs) Transponders convert a client demand to an optical signal with just enough spectrum to carry the demand [33]. The latency incurred on transponders varies depending on their design and supported functionality. More complex transponders include functionality such as in-band management and can have latencies in the range 5 – 10 μs [9]. However, many equipment vendors offer simpler and lower-cost transponders without features such as in-band management. These devices result in transponder processing time L_{txp} as low as 30 ns [26].

FEC processing FEC is used for detecting and correcting transmission errors due to noise and other impairments present in high-capacity transmissions. A stronger FEC increases system margin for a given Bit Error Rate (BER) and optical signal power, thereby increasing the signal-to-noise ratio, enabling longer optical reaches. FEC encoder/decoder modules introduce a processing delay L_{fec} of $\approx$10 μs each for standard FECs [40]. However, processing delay for super FECs with increased correction ability can go up to 150 μs [9]. Since a pair of FEC modules and transponders are involved in a transmission, the delay introduced at the terminal nodes of a lightpath is:

$$L_n = 2(L_{txp} + L_{fec}) \tag{3.1}$$

3.4.2 Latency contributors on a lightpath

A lightpath can span several optical fibers through several bypass EON nodes. Depending on the path's length there can be re-generators, amplifiers, dispersion compensating fibers (DCF), ROADMs and bandwidth-variable optical cross-connect (BV-OXCs) placed along the path, each contributing towards latency. In the following, we discuss the delay introduced by these components.

Fiber propagation The major latency contributor on a lightpath is the propagation delay. Ideally, the speed of light in vacuum (299792.458 km/s) generates a propagation delay of 3.336 μs/km. However, due to optical fiber's refractive index light travels slower in the fiber than in the vacuum, resulting in a propagation delay L_{prop} $\approx$4.9 μs per kilometer of fiber. fiber.

Regenerator 3R (re-amplification, re-shaping, re-timing) regenerators are used to increase the lightpath reach. They also significantly contribute to latency, as 3R regeneration involves optical-to-electrical-to-optical conversion that can take $\approx$100 μs depending on the level of electrical processing needed [9]. In this study, we assume that the substrate EON does not contain any 3R regenerators (rather the modulation format and FEC levels can be tuned to attain different reaches for a transparent lightpath). Therefore, we do not consider any latency incurred by 3R re-generators.

Amplifier Amplifiers are needed to boost the signal strength on long transmission lines [29]. Unlike regenerators, amplifiers operate completely in the optical domain, eliminating the need to separately amplify each individual channel. Erbium doped fiber amplifiers (EDFAs), widely used in long-haul networks, contain up to 30 m long erbium doped fiber that introduces a delay, $L_{amplifier}$, of about 150 ns in each amplifier [26]. One approach to avoid this additional latency is to use expensive Raman amplifiers that use a different optical characteristic to amplify the optical signal. In Raman amplifiers, there are no additional spools of optical fiber and therefore no additional latency is incurred [9]. The number of amplifiers on a lightpath p, $n_{amp}(p)$ is $\approx \lceil len(p)/f_{span} \rceil$, where $len(p)$ is the physical length of p and f_{span} is the typical distance between two amplifiers, called fiber span (in the order of 80 km).

Dispersion compensating fiber Another potential contributor for the latency in a fiber transmission line is the dispersion compensating fiber (DCF) used in long-haul networks [26]. DCF is used to compensate Chromatic Dispersion (CD) of the optical signal that stems from the differential speeds of lightpaths occupying different spectrum ranges in the fibers. The length of DCF is typically between 15% to 25% of the overall fiber length, which essentially increases fiber propagation delay [29]. However, dispersion compensation module (DCM) based on fiber Bragg gratings (FBG) technology can compensate several hundred kilometers of CD without any significant use of additional fiber. DCM keeps the additional latency introduced for CD within a maximum of 50 ns [9]. Nonetheless, in modern coherent transmission systems, CD is mainly compensated through digital signal processing (DSP) at the receiver, eliminating the need for DCF [50]. Therefore, we do not consider any latency pertaining to dispersion compensation in our model.

ROADM and BV-OXC Another component present along a transparent lightpath is the ROADM, installed on the bypass EON nodes. ROADMs may also include BV-OXC that can switch spectrum at intermediate EON nodes. Since optical flows are optically and independently switched by these intermediate devices for different lightpaths, the delays introduced by ROADMs and BV-OXCs, L_{ROADM}, are in the order of tens of ns [36, 53]. The total number of ROADMs (or, BV-OXCs) on a lightpath p is $(|p| + 1)$, where $|p|$ is the number of SLinks on the lightpath.

Based on the above discussion, the latency incurred on a lightpath can be expressed as follows:

$$L_p = L_n + len(p)L_{prop} + n_{amp}(p)L_{amp} + (|p| + 1)L_{roadm} \tag{3.2}$$

In case of VLink embedding with splits (as discussed in Section 3.2), the set of lightpaths supporting a VLink $\bar{e} \in \bar{E}$ can differ with each other in terms of physical length and number of intermediate hops, resulting in different latency for these lightpaths. Hence, the latency of $\bar{e}$ is determined by the lightpath p with the maximum L_p since the destination OTN/FlexE node has to wait for the slowest split before merging all the splits of a VLink. If $\mathcal{P}_{\bar{e}}$ is the set of paths used for embedding the splits of a VLink $\bar{e} \in \bar{E}$, the latency perceived for the VLink $\bar{e} \in \bar{E}$ is as follows:

$$latency(\bar{e}) = \max_{p \in \mathcal{P}_{\bar{e}}} L_p \tag{3.3}$$

3.4.3 Differential delay requirement

VLink embedding by splitting its demand over multiple SPaths imposes additional buffering overhead at the destination OTN/FlexE node to offset the different delays experienced by different splits of the VLink transmitted on different lightpaths. This is also known as the the differential delay problem [13, 24]. To account for differential delay, the destination node needs to store all but the most delayed flow in a buffer until the last flow arrives. The amount of buffer needed to compensate the differential delay depends on both the data-rates of the individual paths and the maximum allowed differential delay (DD_{max}). Both VCAT and FlexE impose very strict bound on DD_{max}, specifically, 250 μs and 10 μs, respectively [1, 13]. We express differential delay requirement for a VLink in terms of the maximum and minimum delay of its embedding SPath set as follows:

$$\forall \bar{e} \in \bar{E} : (\max_{p \in \mathcal{P}_{\bar{e}}} L_p - \min_{p \in \mathcal{P}_{\bar{e}}} L_p) \leq DD_{max} \tag{3.4}$$

3.5 Problem Statement

Given an SN G, a reach table $\mathcal{R}$, and a VN request $\bar{G}$ with VNode mapping function τ and latency constraint set $\mathcal{L}(\bar{G})$:

- Compute the link embedding function $\gamma : \bar{E} \rightarrow \chi : \chi \subset \mathcal{P} \times \mathcal{T} \times S^2$ and $1 \leq |\chi| \leq q$, *i.e.*, compute up to a maximum of q splits for the bandwidth demand $\bar{\beta}_{\bar{e}}$ of each VLink $\bar{e} \in \bar{E}$, select an SPath and an appropriate transmission configuration $t \in \mathcal{T}$ from the reach table $\mathcal{R}$ for each split, and allocate a contiguous segment of slices represented by the starting and ending slice index on each SLink along the SPath. $\chi_{\bar{e}i} = (p, t, s_b, s_t) | 1 \leq i \leq q$ represents the i-th split, where $\chi_{\bar{e}i}^{(p)}$ and $\chi_{\bar{e}i}^{(t)}$ denote the selected SPath and transmission configuration for the i-th split, respectively. In addition, allocation of spectrum slices for the i-th split begins at index $\chi_{\bar{e}i}^{(s_b)}$ and ends at index $\chi_{\bar{e}i}^{(s_t)}$ along each SLink in the SPath $\chi_{\bar{e}i}^{(p)}$. The VLink embedding should be performed in a way such that the latency constraint on virtual paths are satisfied, *i.e.*, $\forall \ell \in \mathcal{L}(\bar{G}) : latency(\ell^{(\bar{a})}) \leq \ell^{(L)}$.
- The total number of slices required to provision the VN is minimum according to the following cost function:

$$\sum_{\forall \bar{e} \in \bar{E}} \sum_{i=1}^{q} (\chi_{\bar{e}i}^{(s_t)} - \chi_{\bar{e}i}^{(s_b)} + 1) \times |\chi_{\bar{e}i}^{(p)}| \tag{3.5}$$

  Here, $|\chi_{\bar{e}i}^{(p)}|$ is the number of SLinks on the SPath $\chi_{\bar{e}i}^{(p)}$.

The above is subject to substrate resource constraints, and spectral contiguity (*i.e.*, the allocated slices of each split are always adjacent to each other) and continuity (*i.e.*, the same sequence of slices are allocated on each SLink along an SPath) constraints on the lightpaths.

3.6 Pre-computations

For each VLink $\bar{e} \in \bar{E}$, we pre-compute $\mathcal{P}_{\bar{e}}^{k}$, a set of k shortest paths between the pair of SNodes where the VLink's endpoints' are mapped. For each SPath $p \in \mathcal{P}_{\bar{e}}^{k}$, we pre-compute the set of admissible transmission configurations, $\mathcal{T}_{\bar{e}p} \subset \mathcal{T}$, such that each configuration $t \in \mathcal{T}_{\bar{e}p}$ results in a reach $r_t \geq len(p)$ and has a data-rate $t^{(d)}$. $\mathcal{T}_{\bar{e}}$ contains all the distinct tuples suitable for $\bar{e}$ and is defined as $\bigcup_{\forall p \in \mathcal{P}_{\bar{e}}^{k}} \mathcal{T}_{\bar{e}p}$.

Chapter 4

Problem Formulation

We present a path-based ILP formulation for optimally solving the problem, *i.e.*, we provide the set of possible SPaths for VLink embedding as input to the ILP. Note that some of the constraints except the latency and differential delay constraints have been presented in different forms in different research works [17, 45, 49]. In the interest of completeness, we present both the common ones and the ones specifically required for the problem at hand.

4.1 Decision Variables

We allow a VLink's bandwidth demand to be satisfied by provisioning slices over multiple SPaths (up to a maximum of q). Slice provisioning over multiple SPaths can be performed using a combination of the following two cases: (i) allocate contiguous sets of slices on distinct SPaths; (ii) allocate different contiguous sets of slices on the same SPath (with the same or different transmission configuration(s)). The latter represents a scenario when there are sufficient slices available over an SPath, however, its length does not allow provisioning the required data-rate using a single contiguous set of slices. To represent the second case, we assume each transmission configuration for an SPath can be instantiated multiple times (up to a maximum of q times). The following variable represents VLink mapping:

$$w_{\bar{e}pti} = \begin{cases} 1 & \text{if } \bar{e} \in \bar{E} \text{ uses } i\text{-th instance of } t \in \mathcal{T}_{\bar{e}p} \\ & \text{on path } p \in \mathcal{P}^k_{\bar{e}} \\ 0 & \text{otherwise} \end{cases}$$

Finally, the following decision variable creates the relationship between a mapped SPath and the slices in its SLinks:

$$y_{\bar{e}ptis} = \begin{cases} 1 & \text{if } \bar{e} \in \bar{E} \text{ uses slice } s \in S \text{ on path } p \in \mathcal{P}_{\bar{e}}^{k} \\ & \text{with the } i\text{-th instance of } t \in \mathcal{T}_{\bar{e}p} \\ 0 & \text{otherwise} \end{cases}$$

4.2 Constraints

4.2.1 VLink demand constraints

We provision a VLink by splitting it across up to q SPaths. Constraint (4.1) ensures that for each VLink $\bar{e} \in \bar{E}$, the sum of data-rates resulting from applying the selected transmission configuration on the selected paths is equal to the VLink's demand. Then, (4.2) enforces an upper limit on the number of splits.

$$\forall \bar{e} \in \bar{E} : \sum_{\forall p \in \mathcal{P}_{\bar{e}}^{k}} \sum_{\forall t \in \mathcal{T}_{\bar{e}p}} \sum_{i=1}^{q} (w_{\bar{e}pti} \times t^{(d)}) = \bar{\beta}_{\bar{e}} \tag{4.1}$$

$$\forall \bar{e} \in \bar{E} : \sum_{\forall p \in \mathcal{P}_{\bar{e}}^{k}} \sum_{\forall t \in \mathcal{T}_{\bar{e}p}} \sum_{i=1}^{q} w_{\bar{e}pti} \leq q \tag{4.2}$$

4.2.2 Slice assignment and Spectral Contiguity constraints

We ensure by (4.3) that if a path p is selected with a specific transmission configuration t, then the required number of slices n_t to support the data-rate $t^{(d)}$ is allocated on the path. (4.4) ensures that each slice on an SLink is allocated to at most one path. Finally, (4.5) ensures the slices allocated on each link of a path form a contiguous frequency spectrum.

$$\forall \bar{e} \in \bar{E}, \forall p \in \mathcal{P}_{\bar{e}}^{k}, \forall t \in \mathcal{T}_{\bar{e}p}, 1 \leq i \leq q : \sum_{\forall s \in S} y_{\bar{e}ptis} = n_t w_{\bar{e}pti} \tag{4.3}$$

$$\forall e \in E, \forall s \in S : \sum_{\forall \bar{e} \in \bar{E}} \sum_{\forall p \in \mathcal{P}_{\bar{e}}^{k}} \sum_{\forall t \in \mathcal{T}_{\bar{e}p}} \sum_{i=1}^{q} w_{\bar{e}pti} y_{\bar{e}ptis} \delta_{pe} \leq 1 \tag{4.4}$$

$$\forall \bar{e} \in \bar{E}, \forall p \in \mathcal{P}_{\bar{e}}^{k}, \forall t \in \mathcal{T}_{\bar{e}p}, 1 \leq i \leq q, 1 \leq s \leq |S| - 1 :$$

$$\sum_{s'=s+2}^{|S|} y_{\bar{e}ptis'} \leq |S| \times (1 - y_{\bar{e}ptis} + y_{\bar{e}pti(s+1)}) \tag{4.5}$$

4.2.3 Latency Constraints

Recall from Section 3.3 that the VN embedding has to satisfy a set of path-based latency constraints $\mathcal{L}(\bar{G})$. Since each such constraint $\ell \in \mathcal{L}(\bar{G})$ involves one or more VLinks, we first introduce a new decision variable $vlink\_lat_{\bar{e}}$ that finds the latency of each VLink $\bar{e}$ using the following linear constraint derived from (3.3):

$$\forall \bar{e} \in \bar{E}, \forall p \in \mathcal{P}_{\bar{e}}^{k}, \forall t \in \mathcal{T}_{\bar{e}p}, 1 \leq i \leq q : L_p w_{\bar{e}pti} \leq vlink\_lat_{\bar{e}} \tag{4.6}$$

The following constraint uses (4.6) to ensure that each latency constraint $\ell \in \mathcal{L}(\bar{G})$ is satisfied:

$$\forall \ell \in \mathcal{L}(\bar{G}) : \sum_{\forall \bar{e} \in \ell^{(\bar{a})}} vlink\_lat_{\bar{e}} \leq \ell^{(L)} \tag{4.7}$$

4.2.4 Differential Delay Constraints

Recall from Section 3.4.3, VLink embedding must satisfy the differential delay requirement (3.4). We use (4.6) and (4.8) to find the maximum and minimum delay of each VLink, respectively. In (4.8), extra care is needed to avoid getting zero as the minimum latency of a VLink induced by a non-selected path, tuple, and instance combination. The second term on the *r.h.s* of (4.8) allows to avoid such cases assuming that λ is sufficiently larger than the maximum value of L_p in the EON. For a non-selected path combination, second term on the *r.h.s* of (4.8) becomes dominant, generating constraint $min\_delay(\bar{e}) \leq \lambda$. For a selected path, tuple, and instance combination, the second term becomes zero and the first term generates the constraint $min\_delay(\bar{e}) \leq L_p$. As $\lambda >> L_p$, $min\_delay(\bar{e}) \leq L_p$ dominates over $min\_delay(\bar{e}) \leq \lambda$ to generate the minimum delay of a VLink as the minimum delay of all the selected path, tuple, and instance combinations. Finally, (4.9) uses (4.6) and (4.8) to enforce the differential delay

requirement for VLinks:

$$\forall \bar{e} \in \bar{E}, \forall p \in \mathcal{P}_{\bar{e}}^{k}, \forall t \in \mathcal{T}_{\bar{e}p}, 1 \leq i \leq q :$$
$$min\_delay(\bar{e}) \leq w_{\bar{e}pti} \times L_p + (1 - w_{\bar{e}pti}) \times \lambda \tag{4.8}$$
$$\forall \bar{e} \in \bar{E} : vlink\_lat_{\bar{e}} - min\_delay(\bar{e}) \leq DD_{max} \tag{4.9}$$

4.3 Objective Function

Our cost function minimizes the total number of spectrum slices required to embed all the VLinks of a VN as shown in the first part of (4.10). However, to break ties among multiple solutions with the same total number of slices, we use the second term in (4.10) that minimizes the number of splits over all the VLinks. This gives us the following objective function:

$$\begin{aligned} \text{minimize}(& \sum_{\forall \bar{e} \in \bar{E}} \sum_{\forall p \in \mathcal{P}_{\bar{e}}^{k}} \sum_{\forall t \in \mathcal{T}_{\bar{e}p}} \sum_{i=1}^{q} \sum_{\forall s \in S} y_{\bar{e}ptis} \times |p| + \\ & \epsilon_1 \times \sum_{\forall \bar{e} \in \bar{E}} \sum_{\forall p \in \mathcal{P}_{\bar{e}}^{k}} \sum_{\forall t \in \mathcal{T}_{\bar{e}p}} \sum_{i=1}^{q} w_{\bar{e}pti}) \end{aligned} \tag{4.10}$$

The weight ϵ_1 in (4.10) is chosen in a way that the second term is effective only when multiple optimal solutions have same value for the first part.

Chapter 5

Heuristic Algorithm

Given the NP-hardness of the optimal solution to the VNE over EON problem even without latency constraints [45], we propose a heuristic to solve large problem instances. In this section, we first give an overview of the embedding process of a VN with latency constraints on an EON (Section 5.1). Then, we discuss how we select the embedding order of VLinks and how we constraint the latency of individual VLinks in a way that satisfies the latency constraints on VPaths (Section 5.2). Finally, we discuss the embedding process of a single VLink with already mapped endpoints (Section 5.3).

5.1 Heuristic solution for VN Embedding

Alg. 1 takes as input a VN $\bar{G}$, an EON G, set of latency constraints $\mathcal{L}(\bar{G})$, and a node mapping function $\tau : \bar{V} \rightarrow V$. One way of computing a cost efficient VLink mapping for a given VNode mapping is to consider all $|\bar{E}|!$ possible orders of sequentially embedding VLinks. The lowest cost mapping among all possible VLink orders then can be chosen as a cost effective solution for the given VNode mapping. However, this brute-force approach is not scalable. Instead, Alg. 1 considers only one sequential VLink embedding order that is dynamically computed to converge to a solution within a reasonable time.

To embed a VLink $\bar{e}$, one needs to compute the maximum latency allowable for $\bar{e}$ that will not violate any of the latency constraints. This is not trivial because $\bar{e}$ can belong to multiple VPaths with different latency constraints, *e.g.*, VLink *qr* on VPaths *p-r-q* and *q-r-s* in Fig. 1.1. What makes it even more difficult is that the mappings (and corresponding latencies) of other VLinks on these VPaths may not be known at this time due to the sequential nature of Alg. 1. For

instance, at the time of estimating the latency budget of the VLink *qr* in Fig. 1.1, the mappings and latency budgets of VLinks *rp* and *rs* might not be known. Therefore, Alg. 1 estimates an upper bound of the latency budget of a VLink $\bar{e}$ based on the latencies of already mapped VLinks, and on the estimated latencies and slice availability in the candidate SPaths of the other unmapped VLinks. Assuming the SPaths in $\mathcal{P}^k_{\bar{e}}$ are sorted in increasing order of their lengths, this upper bound is set to the latency of the maximal i-th SPath $p_i \in \mathcal{P}^k_{\bar{e}} | i \leq k$, such that this chosen value does not violate any of the latency constraints considering the latencies of currently mapped VLinks and the estimated latencies of unmapped ones. Therefore, the estimated latency budget generates an allowed set of candidate SPaths $\mathcal{P}^i_{\bar{e}} = \{p_1, p_2, .., p_i\}$ for $\bar{e}$'s embedding.

Algorithm 1: Algorithm for VN Embedding

```
1  function VNEmbedding(G, Ḡ, L(Ḡ), τ)
2      Ē_remaining ← Ē
3      while Ē_remaining ≠ φ do
4          < ē, i > ← GetNextVLinkToEmbed(G, Ḡ, Ē_remaining, P^k_ē)
5          Z_ē ← FindOptimal(G, ē, P^i_ē, T_ē)
6          foreach e ∈ p | p ∈ I_ē.P^i_ē do
7              Perform slice assignment using I_ē.S
8          χ_ē.P ← I_ē.P^i_ē, χ_ē.T ← I_ē.T_{P^i_ē}
9          if χ_ē = φ then return < φ, φ >
10         Ē_remaining ← Ē_remaining − {ē}
11     return γ : Ē → χ >
```

To increase the chances of finding a feasible solution, Alg. 1 embeds the most constrained VLink $\bar{e}$ in terms of slice availability in its allowed set of candidate SPaths $\mathcal{P}^i_{\bar{e}}$ during each iteration. Alg. 1 uses Alg. 2 to find the most constrained VLink $\bar{e}$ from the set of unmapped VLinks and the allowed set of candidate SPaths $\mathcal{P}^i_{\bar{e}}$ for $\bar{e}$ (details of Alg. 2 will be given in the next section). For the chosen VLink $\bar{e}$, Alg. 1 invokes Alg. 3 to compute the optimal solution for $\bar{e}$ based on $\mathcal{P}^i_{\bar{e}}$ (line 5). Alg. 1 then allocates spectrum slices on all SLinks present in the solution returned by Alg. 3 and updates the VLink mapping function χ. If no solution can be found for any $\bar{e}$, VN embedding fails. Note that after a VLink is mapped, its actual latency can be computed from the set of SPaths used for mapping, and slice availability in the candidate SPaths of the unmapped VLinks will need to be updated. Therefore, Alg. 2 is invoked in each iteration of Alg. 1 to dynamically find the most constrained VLink leveraging the up to date status of embedding and EON.

5.2 VLink Ordering and Latency Budget Allocation

Alg. 2 estimates the latency budgets of the unmapped VLinks in a way such that embedding of VLinks appearing later in the order become more unlikely to fail due to very tight budgets or due to insufficient number of contiguous slices in their candidate SPaths. For instance, the estimated latency budget of the VLink *qr* belonging to VPaths *p-r-q* and *q-r-s* in Fig. 1.1 should not be too large to leave close to zero latency budgets for other VLinks on *p-r-q* or *q-r-s*. One way of ensuring this is to maximize the number of allowed candidate SPaths for the VLinks while satisfying the latency constrains. While this approach works well in a green-field EON with ample spectrum slices, it may lead to infeasible embedding in a dynamic environment where some of the SPaths in $\mathcal{P}^i_{\bar{e}}$ are fragmented due to accommodating existing VNs. Therefore, Alg. 2 takes slice availability in the candidate SPaths into account while estimating latency budgets.

Alg. 2 computes an index $i_{\bar{e}}$ for each VLink $\bar{e}$, denoting the maximum index of the SPath in the set of allowed SPaths for $\bar{e}$ corresponding to the latency budget. While computing this index, Alg. 2 employs spectrum awareness by maximizing the minimum number of usable slices in the allowed set of SPaths for each of the remaining VLinks while satisfying all latency constraints. To do so, Alg. 2 performs binary search over the interval between 1 and the minimum value of the number of free slices in the candidate SPaths of the remaining VLinks. In each iteration of binary search, Alg. 2 chooses the median value med in the range and increases $i_{\bar{e}}$ until the total number of free slices in the SPaths in $\mathcal{P}^i_{\bar{e}}$ becomes equal or more than med. It then uses $i_{\bar{e}}$ to compute the estimated latency of $\bar{e}$. After estimating latencies for all the unmapped VLinks, Alg. 2 checks if any latency constraint in $\mathcal{L}(\bar{G})$ is violated. Based on this test, the binary search continues in the appropriate range and terminates when the range size becomes 1.

The binary search in Alg. 2 provides a lower bound a on the minimum number of usable slices across all the VLinks. However, it does not identify the VLink(s) with the minimum number of usable slices. According to the property of binary search, there exists at least one VLink for which $i_{\bar{e}}$ cannot be increased beyond a without violating at least one latency constraint. Alg. 2 identifies such VLink by attempting to increase the $i_{\bar{e}}$ for each virtual link $\bar{e}$ and checking if all the latency constraints can still be satisfied using the expanded set of allowed SPaths. The VLink that fails this test is then returned as the most constrained one along with its corresponding index $i_{\bar{e}}$ (line 28 – 29). If multiple such VLinks exist, Alg. 2 returns the one with higher demand, which is ensured by traversing the VLinks in decreasing order of their demand (line 23).

Algorithm 2: Algorithm for finding a VLink based on latency constraints and spectrum availability

function *GetNextVLinkToEmbed*$(G, \bar{G}, \bar{E}_{rest}, \mathcal{P}^k_{\bar{e}}, \mathcal{L}(\bar{G}))$
  foreach $\bar{e} \in \bar{E} \setminus \bar{E}_{rest}$ **do**
    $p \leftarrow$ SPath with the longest length in $\chi_{\bar{e}}.\mathcal{P}$
    $est\_latency(\bar{e}) \leftarrow L_p$ using (3.2)
  $\mathcal{L} \leftarrow$ all latency constraints in $\mathcal{L}(\bar{G})$
  $a \leftarrow 1, b \leftarrow \min_{\bar{e} \in \bar{E}_{rest}} \sum_{p_i \in \mathcal{P}^k_{\bar{e}}} free\_slices(p_i)$
  while $b - a > 1$ **do**
    $med \leftarrow \frac{a+b}{2}$
    foreach $\bar{e} \in \bar{E}_{rest}$ **do**
      $i_{\bar{e}} \leftarrow free(\bar{e}) \leftarrow 0$
      while $free(\bar{e}) < med$ **do**
        $i_{\bar{e}} \leftarrow i_{\bar{e}} + 1$
        $free(\bar{e}) \leftarrow free(\bar{e}) + free\_slices(p_{i_{\bar{e}}})$
      $est\_latency(\bar{e}) \leftarrow L_{p_{i_{\bar{e}}}}$ using (3.2)
    all_constraints_satisfied$\leftarrow true$
    foreach $l \in \mathcal{L}$ **do**
      $est\_latency(\ell^{(\bar{a})}) = \sum_{\bar{e} \in \bar{a}} est\_latency(\bar{e})$
      if $est\_latency(\ell^{(\bar{a})}) > \ell^{(L)}$ **then**
        all_constraints_satisfied$\leftarrow false$
        break
    if *all_constraints_satisfied* $= true$ **then** $a \leftarrow med$
    else $b \leftarrow med$
  foreach $\bar{e} \in \bar{E}_{rest}$ *in decreasing order of* $\bar{\beta}_{\bar{e}}$ **do**
    $i_{\bar{e}} \leftarrow i_{\bar{e}} + 1$
    $est\_latency(\bar{e}) \leftarrow L_{p_{i_{\bar{e}}}}$ using (3.2)
    foreach $l \in \mathcal{L}$ **do**
      $est\_latency(\ell^{(\bar{a})}) = \sum_{\bar{e} \in \bar{a}} est\_latency(\bar{e})$
      if $est\_latency(\ell^{(\bar{a})}) > \ell^{(L)}$ *and* $free(\bar{e}) = a$ **then**
        return $< \bar{e}, i_{\bar{e}} - 1 >$

5.2.1 Running Time Analysis

Alg. 2 performs a binary search on a range of $k \times |S|$, resulting in $O(\log k|S|)$ iterations. In each iteration of the search, latency budget estimation can take $O(k|\bar{E}|)$ time since we need to check

all the VLinks and all their k candidate shortest paths at the initial stage. Each iteration of the binary search also involves checking $|\mathcal{L}|$ latency constraints, each of which can have $|\bar{V}|$ VLinks, requiring $O(|\mathcal{L}||\bar{V}|)$ time. Finally, finding the most constrained VLink takes $|\bar{E}| \times |\mathcal{L}||\bar{V}|$ time as $|\mathcal{L}|$ latency constraints need to be checked for each VLink. Therefore, the total running time of Alg. 2 is $O(\log(k|S|) \times (k|\bar{E}| + |\mathcal{L}||\bar{V}|) + |\bar{E}| \times |\mathcal{L}||\bar{V}|)$.

5.3 Optimal solution for a single VLink with Mapped Endpoints

Once we have a VLink with mapped endpoints and a set of candidate shortest SPaths defined by the latency constraints, embedding the VLink on an EON involves the following steps:

Path Selection Recall from Section 3.2 that a VLink's demand can be satisfied by dividing it into maximum q splits and provisioning each split on an SPath in the EON. Note that different contiguous segments of spectrum slices on the same SPath can be allocated to provision multiple splits on the same SPath. Therefore, the choice of up to q paths from the set of candidate shortest SPaths (which can be up to k) between a VLink's mapped endpoints for embedding a VLink $\bar{e}$ can be represented using a multi-set $\mathbb{P}^k_{\bar{e}} = (\mathcal{P}^k_{\bar{e}}, m_1)$. $m_1 : \mathcal{P}^k_{\bar{e}} \rightarrow N_q$ (N_q is the set of all integers between 0 and q inclusive) defines the number of times a path $p \in \mathcal{P}^k_{\bar{e}}$ appears in the multi-set $\mathbb{P}^k_{\bar{e}}$. $m_1(p) = 0$ indicates exclusion of a path $p \in \mathcal{P}^k_{\bar{e}}$ from $\mathbb{P}^k_{\bar{e}}$. Size of $\mathbb{P}^k_{\bar{e}}$ is then computed as, $|\mathbb{P}^k_{\bar{e}}| = \sum_{p \in \mathcal{P}^k_{\bar{e}}} m_1(p)$, where $1 \leq |\mathbb{P}^k_{\bar{e}}| \leq q$. Since order does not matter in discriminating multisets, $\mathrm{P}^k_{\bar{e}}$ represents a permutation of the SPaths present in the multi-set $\mathbb{P}^k_{\bar{e}}$.

Transmission Configuration Selection Each SPath $p \in \mathbb{P}^k_{\bar{e}}$ with $m_1(p) \geq 1$ can be configured with different transmission configurations to provision different data-rates in such a way that altogether they achieve the VLink's demand $\beta_{\bar{e}}$. Recall from Section 4.1 that the same transmission configuration can be applied to the same path or different paths multiple times to allocate different contiguous segments of spectrum slices. Therefore, selection of transmission configurations for the paths in $\mathbb{P}^k_{\bar{e}}$ results in a multi-set, $\mathbb{T}_{\mathbb{P}^k_{\bar{e}}} = (\mathcal{T}_{\mathbb{P}^k_{\bar{e}}}, m_2)$, where $\mathcal{T}_{\mathbb{P}^k_{\bar{e}}}$ is the set of feasible transmission configurations for the paths in $\mathbb{P}^k_{\bar{e}}$ and m_2 defines the number of times a transmission configuration appears in $\mathbb{T}_{\mathbb{P}^k_{\bar{e}}}$. There is an one-to-one correspondence between a path in $\mathbb{P}^k_{\bar{e}}$ and a selected transmission configuration in $\mathbb{T}^k_{\mathbb{P}_{\bar{e}}}$, hence, $|\mathbb{T}_{\mathbb{P}^k_{\bar{e}}}| = |\mathbb{P}^k_{\bar{e}}|$.

Spectrum Slice Allocation The final step in VLink mapping is to allocate contiguous segment of spectrum slices on the SPaths in $\mathbb{P}^k_{\bar{e}}$, *i.e.*, allocate identical set of spectrum slice indices on each SLink along each SPath $p \in \mathbb{P}^k_{\bar{e}}$. The number of spectrum slices to allocate is determined by the transmission configuration $t \in \mathbb{T}_{\mathbb{P}^k_{\bar{e}}}$ selected for p. Spectrum slice allocation for a pair of $< p, t >$ is represented by the starting slice index $s_b \in S$. The ending slice index for each such pair can be trivially derived using $s_t = s_b + n_t - 1$, where n_t is the number slices required for t. Since more than one distinct SPaths in $\mathbb{P}^k_{\bar{e}}$ can have the same starting slice index, the assignment of spectrum slices to the SPaths in $\mathbb{P}^k_{\bar{e}}$ can be represented as a multi-set of slice indices, $\mathbb{S}_{\mathbb{P}^k_{\bar{e}}} = (S, m_3)$ such that $|\mathbb{S}_{\mathbb{P}^k_{\bar{e}}}| = |\mathbb{P}^k_{\bar{e}}|$, satisfying constraints (4.4) and (4.5). $m_3 : S_{\mathbb{P}^k_{\bar{e}}} \rightarrow N_q$ defines the number of times a slice index from S appears in the multi-set $\mathbb{S}_{\mathbb{P}^k_{\bar{e}}}$. Again, there is an one-to-one correspondence between an SPath in $\mathbb{P}^k_{\bar{e}}$ and a slice assignment in $\mathbb{S}_{\mathbb{P}^k_{\bar{e}}}$.

In summary, embedding solution of a VLink $\bar{e}$, $\mathcal{I}_{\bar{e}}$, can be represented as a tuple of three multi-sets, $\mathcal{I}_{\bar{e}} = (\mathbb{P}^k_{\bar{e}}, \mathbb{T}_{\mathbb{P}^k_{\bar{e}}}, \mathbb{S}_{\mathbb{P}^k_{\bar{e}}})$. The optimal embedding for $\bar{e}$, $\mathcal{A}_{\bar{e}}$, is the one requiring minimum number of spectrum slices. If $\mathcal{M}(\mathcal{P}^k_{\bar{e}})$, $\mathcal{M}(\mathcal{T}_{\mathbb{P}^k_{\bar{e}}})$, and $\mathcal{M}(\mathcal{S}_{\mathbb{P}^k_{\bar{e}}})$ are the set of all possible multi-sets of $\mathcal{P}^k_{\bar{e}}$, $\mathcal{T}_{\mathbb{P}^k_{\bar{e}}}$, and $\mathcal{S}_{\mathbb{P}^k_{\bar{e}}}$, respectively, and $n(\mathcal{I}_{\bar{e}})$ is the number of slices used by solution $\mathcal{I}_{\bar{e}}$, then $\mathcal{A}_{\bar{e}}$ can be defined as:

$$\mathcal{A}_{\bar{e}} = \underset{\mathcal{I}_{\bar{e}} \in \mathcal{M}(\mathbb{P}^k_{\bar{e}}) \times \mathcal{M}(\mathbb{T}_{\mathbb{P}^k_{\bar{e}}}) \times \mathcal{M}(\mathbb{S}_{\mathbb{P}^k_{\bar{e}}})}{\arg\min} n(\mathcal{I}_{\bar{e}}) \tag{5.1}$$

Our proposed heuristic for a single VLink embedding with already mapped endpoints (Alg. 3) enumerates all such tuples in a brute-force manner and selects the one that uses the minimum number of spectrum slices. To accomplish this, we first enumerate all possible multi-sets $\mathbb{P}^k_{\bar{e}}$ of sizes ranging from 1 to q, *i.e.*, all possible selection of up to q paths from $\mathcal{P}^k_{\bar{e}}$. For each enumerated $\mathbb{P}^k_{\bar{e}} \in \mathcal{M}(\mathbb{P}^k_{\bar{e}})$, Alg. 3 needs to compute $\mathbb{T}^{opt}_{\mathbb{P}^k_{\bar{e}}}$ and $\mathbb{S}^{opt}_{\mathbb{P}^k_{\bar{e}}}$, the multi-sets of transmission configuration and slice indices, respectively, which minimizes $n(\mathbb{P}^k_{\bar{e}})$, the number of spectrum slices required for $\mathbb{P}^k_{\bar{e}}$. However, while computing $\mathbb{T}^{opt}_{\mathbb{P}^k_{\bar{e}}}$ and $\mathbb{S}^{opt}_{\mathbb{P}^k_{\bar{e}}}$ Alg. 3 avoids enumerating all possible multi-sets by using the following technique.

For a path selection $\mathbb{P}^k_{\bar{e}}$, Alg. 3 first check whether $\mathbb{P}^k_{\bar{e}}$ satisfies the required differential delay constraint using the following equation:

$$\begin{aligned} &\forall (p_1, p_2) \in (\mathcal{P}^k_{\bar{e}} \times \mathcal{P}^k_{\bar{e}}) s.t. p_1 \neq p_2 : \\ &diff\_delay(\mathbb{P}^k_{\bar{e}}) = |L_{p_1} - L_{p_2}| \leq max\_diff\_delay(\bar{e}) \end{aligned} \tag{5.2}$$

Then it uses the reach table to enumerate the possible combinations of $|\mathbb{P}^k_{\bar{e}}|$ data-rates such that their sum equals the VLink's demand $\beta_{\bar{e}}$. These combinations of data-rates is represented by a

Algorithm 3: Find the optimal solution for a single VLink

1 **function** *FindOptimal*$(G, \bar{e}, \mathcal{P}^k_{\bar{e}}, \mathcal{T}_{\bar{e}})$
2 $\quad \mathcal{M}(\mathbb{P}^k_{\bar{e}}) \leftarrow \phi$
3 $\quad$ **for** $i \in N_q$ **do**
4 $\quad\quad \mathcal{M}(\mathbb{P}^k_{\bar{e}}) \leftarrow \mathcal{M}(\mathbb{P}^k_{\bar{e}}) \cup$ Multi-Set$(\mathcal{P}^k_{\bar{e}}, i)$
5 $\quad$ **foreach** $\mathbb{P}^k_{\bar{e}} \in \mathcal{M}(\mathbb{P}^k_{\bar{e}})$ **do**
6 $\quad\quad$ **if** $\mathbb{P}^k_{\bar{e}}$ *does not satisfy* ***differential-delay-constraint*** *(5.2)* **then**
7 $\quad\quad\quad$ **continue**
8 $\quad\quad \mathcal{M}(\mathbb{D}_{\mathbb{P}^k_{\bar{e}}}) \leftarrow$ Multi-Set$(\mathcal{D}_{\mathbb{P}^k_{\bar{e}}}, |\mathbb{P}^k_{\bar{e}}|)$ s.t. $\sum_{d \in \mathbb{D}_{\mathbb{P}^k_{\bar{e}}}} d = \bar{\beta}_{\bar{e}}$
9 $\quad\quad$ **foreach** $\mathbb{D}_{\mathbb{P}^k_{\bar{e}}} \in \mathcal{M}(\mathbb{D}_{\mathbb{P}^k_{\bar{e}}})$ **do**
10 $\quad\quad\quad \zeta(\mathbb{D}_{\mathbb{P}^k_{\bar{e}}}) \leftarrow$ AllPerm$(\mathbb{D}_{\mathbb{P}^k_{\bar{e}}})$
11 $\quad\quad\quad$ **foreach** $\mathrm{D}_{\mathbb{P}^k_{\bar{e}}} \in \zeta(\mathbb{D}_{\mathbb{P}^k_{\bar{e}}})$ **do**
12 $\quad\quad\quad\quad < n, \mathbb{P}, \mathbb{T}, \mathbb{S} > \leftarrow$ MDP$(\mathbb{P}^k_{\bar{e}}, \mathrm{D}_{\mathbb{P}^k_{\bar{e}}})$
13 $\quad\quad$ Find $\mathbb{P}^{opt}$, $\mathbb{T}^{opt}$ and $\mathbb{S}^{opt}$ that minimizes $n(\mathbb{P}^k_{\bar{e}}, \mathrm{D}_{\mathbb{P}^k_{\bar{e}}})$ using (5.3)
14 $\quad\quad \mathcal{I}_{\bar{e}} \leftarrow < \mathbb{P}^{opt}, \mathbb{T}^{opt}, \mathbb{S}^{opt} >$
15 $\quad\quad \mathtt{Y}_{\bar{e}} \leftarrow \mathtt{Y}_{\bar{e}} \cup \mathcal{I}_{\bar{e}}$
16 $\quad$ **return** $\mathtt{Z}_{\bar{e}} \in \mathtt{Y}_{\bar{e}}$ as the best solution that minimizes (5.1)

multi-set $\mathbb{D}_{\mathbb{P}^k_{\bar{e}}} = (\mathcal{D}_{\mathbb{P}^k_{\bar{e}}}, m_4)$, where $\mathcal{D}_{\mathbb{P}^k_{\bar{e}}}$ is the set of all feasible data-rates for the paths in $\mathbb{P}^k_{\bar{e}}$ according to the reach table, and $m_4 : \mathcal{D}_{\mathbb{P}^k_{\bar{e}}} \rightarrow N_q$ defines the number of times a data-rate appears in $\mathbb{D}_{\mathbb{P}^k_{\bar{e}}}$. Note that $|\mathbb{D}_{\mathbb{P}^k_{\bar{e}}}| = |\mathbb{P}^k_{\bar{e}}|$ and $\sum_{d \in \mathbb{D}_{\mathbb{P}^k_{\bar{e}}}} d = \bar{\beta}_{\bar{e}}$.

Provisioning the same data-rate over different paths requires different number of spectrum slices since the paths differ in both physical distance and hop count. To take this into account, for each $\mathbb{D}_{\mathbb{P}^k_{\bar{e}}}$, Alg. 3 considers all permutations of data-rates from $\mathbb{D}_{\mathbb{P}^k_{\bar{e}}}$ (denoted by $\zeta(\mathbb{D}_{\mathbb{P}^k_{\bar{e}}})$) assigned to the paths in $\mathbb{P}^k_{\bar{e}}$ and computes the number of required spectrum slices for each assignment using (5.3) (lines 10 – 13).

$$n(\mathbb{P}^k_{\bar{e}}) = \min_{\forall \mathrm{D}_{\mathbb{P}^k_{\bar{e}}} \in \zeta(\mathbb{D}_{\mathbb{P}^k_{\bar{e}}}) | \forall \mathbb{D}_{\mathbb{P}^k_{\bar{e}}} \in \mathcal{M}(\mathbb{D}_{\mathbb{P}^k_{\bar{e}}})} n(\mathbb{P}^k_{\bar{e}}, \mathrm{D}_{\mathbb{P}^k_{\bar{e}}}) \tag{5.3}$$

Here, $n(\mathbb{P}^k_{\bar{e}}, \mathrm{D}_{\mathbb{P}^k_{\bar{e}}})$ is the number of slices required for a $\mathrm{D}_{\mathbb{P}^k_{\bar{e}}} \in \zeta(\mathbb{D}_{\mathbb{P}^k_{\bar{e}}})$ and $\mathcal{M}(\mathbb{D}_{\mathbb{P}^k_{\bar{e}}})$ represents all possible multi-sets of $\mathbb{D}_{\mathbb{P}^k_{\bar{e}}}$. $\mathbb{T}^{opt}_{\mathbb{P}^k_{\bar{e}}}$ and $\mathbb{S}^{opt}_{\mathbb{P}^k_{\bar{e}}}$ are then the transmission configuration selection and spectrum allocation to provision the data-rates to path assignment, respectively, resulting in the the minimum number of spectrum slices.

A key challenge here is to efficiently evaluate (5.3) without computing for all possibilities.

Algorithm 4: Algorithm for Computing $< n, \mathbb{P}, \mathbb{T}, \mathbb{S} >$ for $\mathbb{P}^k_{\bar{e}}$ and $\mathrm{D}_{\mathbb{P}^k_{\bar{e}}}$

function *MDP(*$\mathbb{P}^k_{\bar{e}}, \mathrm{D}_{\mathbb{P}^k_{\bar{e}}}$*)*
  if $F[\mathbb{P}^k_{\bar{e}}, \mathrm{D}_{\mathbb{P}^k_{\bar{e}}}] \neq \phi$ **then**
    return $F[\mathbb{P}^k_{\bar{e}}, \mathrm{D}_{\mathbb{P}^k_{\bar{e}}}]$
  if $|\mathbb{P}^k_{\bar{e}}| = 1$ **then**
    Find t that minimizes n_t for $p_1 \in \mathbb{P}^k_{\bar{e}}$ and $d_1 \in \mathrm{D}_{\mathbb{P}^k_{\bar{e}}}$ using (5.5)
    if $t = \phi$ *or First-Fit*$(p_1, n_t) = \phi$ **then**
      $n \leftarrow \infty$, $\mathcal{P} \leftarrow \mathcal{T} \leftarrow \mathcal{S} \leftarrow \phi$
    else
      $n \leftarrow n_t$, $\mathcal{P} \leftarrow \{p_1\}$
      $\mathcal{T} \leftarrow \{t\}$, $\mathcal{S} \leftarrow$ First-Fit(p_1, n_t)
    Store $< n, \mathcal{P}, \mathcal{T}, \mathcal{S} >$ to F and return
  else
    for $i = 1$ *to* $|\mathbb{P}^k_{\bar{e}}|$ **do**
      $< n_i, \mathcal{P}_i, \mathcal{T}_i, \mathcal{S}_i > \leftarrow$ MDP$(\{p_i\}, \{d_i\})$
    $n \leftarrow \sum_{i=1}^{i=|\mathbb{P}^k_{\bar{e}}|} n_i$, $\mathcal{P} \leftarrow \bigcup_{i=1}^{i=|\mathbb{P}^k_{\bar{e}}|} \mathcal{P}_i$
    $\mathcal{T} \leftarrow \bigcup_{i=1}^{i=|\mathbb{P}^k_{\bar{e}}|} \mathcal{T}_i$, $\mathcal{S} \leftarrow \bigcup_{i=1}^{i=|\mathbb{P}^k_{\bar{e}}|} \mathcal{S}_i$
    if $n = \infty$ **then**
      Store $< \infty, \phi, \phi, \phi >$ to F and return
    else if *All paths in* $\mathbb{P}^k_{\bar{e}}$ *are disjoint* **then**
      Store $< n, \mathcal{P}, \mathcal{T}, \mathcal{S} >$ to F and return
    else
      $\zeta(\mathbb{P}^k_{\bar{e}}) \leftarrow$ AllPerm$(\mathbb{P}^k_{\bar{e}})$
      foreach $\mathrm{P}^k_{\bar{e}} \in \zeta(\mathbb{P}^k_{\bar{e}})$ **do**
        $\mathcal{S}_{new} \leftarrow \infty$
        foreach $p_i \in \mathrm{P}^k_{\bar{e}}$ **do**
          $\mathcal{S}_i \leftarrow$ First-Fit$(p_i, n(\{p_i\}, \{t_i^{(d)}\}))$
          if $\mathcal{S}_i = \infty$ **then**
            $\mathcal{S}_{new} \leftarrow \infty$
            break
          $\mathcal{S}_{new} \leftarrow \mathcal{S}_{new} \bigcup \mathcal{S}_i$
        if $\mathcal{S}_{new} \neq \infty$ **then**
          Store $< n, \mathrm{P}^k_{\bar{e}}, \mathcal{T}, \mathcal{S}_{new} >$ to F and return
      if *There is no valid slice assignment* **then**
        Store $< \infty, \phi, \phi, \phi >$ to F and return

We observe that computation of the optimal $n(\mathbb{P}^k_{\bar{e}}, \mathrm{D}_{\mathbb{P}^k_{\bar{e}}})$ exhibits optimal substructures and overlapping sub-problems, which allow us to leverage memoized Dynamic Programming (DP) to efficiently evaluate (5.3). Alg. 3 invokes a procedure called MDP (Alg. 4), which uses DP to

compute $n(\mathbb{P}^k_{\bar{e}}, \mathrm{D}_{\mathbb{P}^k_{\bar{e}}})$ for a data-rate permutation $\mathrm{D}_{\mathbb{P}^k_{\bar{e}}} \in \zeta(\mathbb{D}_{\mathbb{P}^k_{\bar{e}}})$ assigned to a path selection $\mathbb{P}^k_{\bar{e}}$ as follows.

Given $\mathrm{D}_{\mathbb{P}^k_{\bar{e}}} = \langle d_1, d_2, .., d_{|\mathbb{P}^k_{\bar{e}}|} \rangle$, an ordered set representing a permutation of $\mathbb{D}_{\mathbb{P}^k_{\bar{e}}}$, Alg. 4 divides the problem into $|\mathbb{P}^k_{\bar{e}}|$ sub-problems. Each of these sub-problems is represented as $(\{p_i\}, \{d_i\})$, where d_i is the i-th datarate in $\mathrm{D}_{\mathbb{P}^k_{\bar{e}}}$ and p_i is the i-th path in $\mathbb{P}^k_{\bar{e}}$ (Line - 14). Each $d_i \in \mathrm{D}_{\mathbb{P}^k_{\bar{e}}} | m_4(d_i) > 1$ may result in multiple occurrences of the same sub-problems for paths $p_i \in \mathbb{P}^k_{\bar{e}}$ with $m_1(p_i) > 1$ (*i.e.*, overlapping sub-problems). These sub-problems also have optimal sub-structure property since the optimal solution for $n(\mathbb{P}^k_{\bar{e}}, \mathrm{D}_{\mathbb{P}^k_{\bar{e}}})$ can be computed using the following recurrence:

$$n(\mathbb{P}^k_{\bar{e}}, \mathrm{D}_{\mathbb{P}^k_{\bar{e}}}) = \sum_{i=1}^{i=|\mathbb{P}^k_{\bar{e}}|} n(\{p_i\}, \{d_i\}) \tag{5.4}$$

These two properties allow us to memoize results from already solved sub-problems (line 25) and use them for subsequent recursive invocations of Alg. 4 (lines 2 – 3).

Alg. 4 uses (5.5) to find the most spectrally efficient transmission configuration t such that $t^{(d)} = d_i$ and t results in a transmission reach of at least the length of p_i

$$n(\{p_i\}, \{d_i\}) = \begin{cases} \min_{\forall t \in \mathcal{T}_{\bar{e}p_i} | t^{(d)} = d_i \wedge r_t \geq len(p_i)} n_t \\ \infty \text{ if no } t \text{ can be found} \end{cases} \tag{5.5}$$

In addition to $n(\{p_i\}, \{d_i\})$, Alg. 4 also returns the corresponding transmission configuration t and a slice assignment $\mathcal{S}$ consisting of contiguous and free n_t slices for p_i. Currently, we are using a First-fit approach to find $\mathcal{S}$ [12]. Note that it is possible that the SLinks on SPath p_i do not have n_t contiguous slices, rendering the sub-problem infeasible to solve (lines 17 – 18 and 26 – 27).

Alg. 4 combines the transmission configurations returned by the sub-problems to obtain a valid transmission configuration selection $\mathcal{T}$ for the original problem. However, combining slice index selections $\mathcal{S}$ requires extra care for the following reasons. First, spectrum resources are shared among the SPaths in $\mathbb{P}^k_{\bar{e}}$ with common SLinks. Second, sub-problems are solved independently without considering the shared spectrum resources on the common SLinks between the SPaths. As a result, slice index selection, $\mathcal{S}$ of two sub-problems containing two paths with a common SLink can end up using the same spectrum slice. Therefore, these slice selections may become invalid when merging the solutions for the sub-problems for violating the requirement for non-overlapping spectrum. This also holds for the solution to the sub-problems involving the

same paths. However, for the paths in $\mathbb{P}^k_{\bar{e}}$ that are disjoint, their slice index assignments from the sub-problems can be readily combined (line 20).

For the paths in $\mathbb{P}^k_{\bar{e}}$ that are not disjoint, Alg. 4 has to select $n(\{p_i\}, \{t_i^{(d)}\})$ contiguous and un-occupied slices along each SPath $p_i \in \mathbb{P}^k_{\bar{e}}$. The order in which Alg. 4 assigns slices to these paths has an impact on the subsequent slice selection. For instance, slice selections to $|\mathbb{P}^k_{\bar{e}}| - 1$ paths may fragment the spectrum on one of the SPaths in $\mathbb{P}^k_{\bar{e}}$ in such a way that the SPath does not have the required number of contiguous and un-occupied slices for the final slice assignment. To circumvent this issue, Alg. 4 enumerates all permutations of the SPaths in $\mathbb{P}^k_{\bar{e}}$ (denoted by $\zeta(\mathbb{P}^k_{\bar{e}})$) to find a feasible slice selection. For a given permutation $\mathrm{P}^k_{\bar{e}} \in \zeta(\mathbb{P}^k_{\bar{e}})$, it selects first available $n(\{p_i\}, \{t_i^{(d)}\})$ contiguous slices on each path $p_i \in \mathrm{P}^k_{\bar{e}}$ in the order p_i appears in $\mathrm{P}^k_{\bar{e}}$. However, Alg. 4 may not find any $\mathcal{S}_{new}$ even after exploring all permutations in $\zeta(\mathbb{P}^k_{\bar{e}})$. The sub-problem is considered infeasible for such cases (Line 34).

5.3.1 Running Time Analysis

Alg. 3 explores all the possibilities of choosing the multi-set $\mathbb{P}^k_{\bar{e}}$. The number of multi-sets of cardinality q on ground set of k SPaths is termed as k multichoose q and is given by $\left(\!\binom{k}{q}\!\right) = \binom{k+q-1}{q}$ [39]. Since $|\mathbb{P}^k_{\bar{e}}|$ can take any integer value from 1 to q, we get $|\mathcal{M}(\mathbb{P}^k_{\bar{e}})| = \sum_{i=1}^{i=q} \left(\!\binom{k}{i}\!\right)$. For each multi-set $\mathbb{P}^k_{\bar{e}}$, Alg. 3 enumerates all possible multisets $\mathbb{D}_{\mathbb{P}^k_{\bar{e}}} \in \mathcal{M}(\mathbb{D}_{\mathbb{P}^k_{\bar{e}}})$, resulting in $\binom{|\mathcal{D}_{\mathbb{P}^k_{\bar{e}}}|+q-1}{q}$ enumerations in the worst case. Alg. 3 invokes Alg. 4 for all the permutations of each of the multisets $\mathbb{D}_{\mathbb{P}^k_{\bar{e}}} \in \mathcal{M}(\mathbb{D}_{\mathbb{P}^k_{\bar{e}}})$. The number of permutations of a multiset $\mathbb{D}_{\mathbb{P}^k_{\bar{e}}}$ of cardinality q is given by the multinomial coefficient, *i.e.*, $\frac{q!}{\Pi_{d_j \in \mathcal{D}_{\mathbb{P}^k_{\bar{e}}}} m_4(d_j)!}$ [11]. Therefore, Alg. 3 invokes Alg. 4 $\frac{(|\mathcal{D}_{\mathbb{P}^k_{\bar{e}}}|+q-1)!}{(|\mathcal{D}_{\mathbb{P}^k_{\bar{e}}}|-1)! \times \Pi_{d_j \in \mathcal{D}_{\mathbb{P}^k_{\bar{e}}}} m_4(d_j)!}$ times to compute $n(\mathbb{P}^k_{\bar{e}})$. The most expensive step of Alg. 4 is the exploration of all the permutations of the paths in $\mathbb{P}^k_{\bar{e}}$ requiring $\frac{q!}{\Pi_{p_j \in \mathcal{P}^k_{\bar{e}}} m_1(p_j)!}$ possibilities in the worst case. Therefore, to find $\mathcal{A}_{\bar{e}}$, Alg. 3 enumerates $\left(\sum_{i=1}^{i=q} \binom{k+i-1}{i}\right) \times \frac{(|\mathcal{D}_{\mathbb{P}^k_{\bar{e}}}|+q-1)!}{(|\mathcal{D}_{\mathbb{P}^k_{\bar{e}}}|-1)! \times \Pi_{d_j \in \mathcal{D}_{\mathbb{P}^k_{\bar{e}}}} m_4(d_j)!} \times \frac{q!}{\Pi_{p_j \in \mathcal{P}^k_{\bar{e}}} m_1(p_j)!}$ possibilities. Typical values of k and q are small, therefore, the running time is dominated by the size of $\mathcal{D}_{\mathbb{P}^k_{\bar{e}}}$.

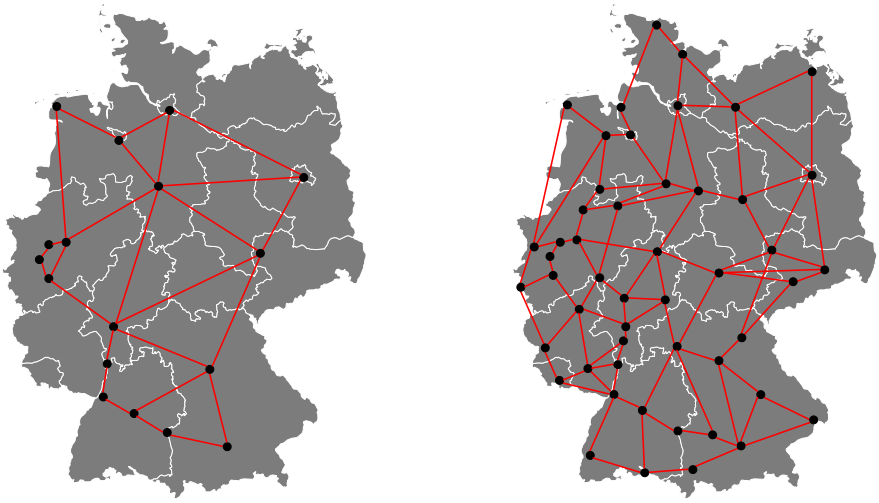
Chapter 6

Evaluation

6.1 Simulation Setup

6.1.1 Testbed and Topology

We have implemented both the ILP formulation and the heuristic algorithm in C++. We use Nobel Germany, Fig. 6.1(a), (17 nodes and 26 links) and Germany50, Fig. 6.1(b), (50 nodes and 88 links) networks from SNDlib repository [2] as the SNs for small and large scale simulations, respectively. Ten shortest paths between all pairs of SNodes in each SN are pre-computed as input. Spectrum bandwidth on each SLink is set to 600GHz and 4THz for small scale and large scale simulations, respectively. For the steady state analysis, we use the Nobel Germany SN and set 4THz spectrum bandwidth on the SLinks. Depending on the evaluation scenario, we consider fixed- and flex-grid variations of the SN. The fixed-grid variation allocates spectrum slices in 50Ghz granularity and considers only a few data rates supported by the transponders (*i.e.*, 100Gbps, 200Gbps, and 400Gbps). Whereas, the flex-grid case allocates spectrum in 12.5Ghz granularity and allows transponders to support a higher number of data rates up to 800Gbps. To achieve a given data rate, a transponder is allowed to choose from a number of transmission configurations provided in reach tables (as discussed in Section 3.1) for both fixed- and flex-grid variations. Latency characteristics of the SNs are set according to Table 3.1. We synthetically generate the VNs with different properties based on the simulation scenario. We map each VNode to a random SNode while ensuring that no two VNodes from the same VN are mapped to the same SNode. VLink demands are varied between 100Gbps to 1Tbps with possible values as multiples of 100Gbps. Simulations are performed on a machine with 8×10-core 2.40GHz Intel Xeon E7-8870 processors and 1TB memory.



(a) Nobel Germany network (b) Germany50 network

Figure 6.1: Substrate networks are used for evaluation

6.1.2 VN generation for microbenchmarking

In this scenario, we consider each VN request in isolation for each of the compared approaches discussed in Section 6.1.5. We vary the VNs by changing their link to node ratio (LNR) and the total bandwidth demand, while keeping the number of VNodes fixed at 8 and 50 for small- and large-scale simulations, respectively. For each simulation run, we generate 5 and 50 VNs with the same LNR and similar total bandwidth demands for small- and large-scale simulations, respectively, and report the mean of the performance metrics over those VNs.

6.1.3 VN generation for steady state analysis

This analysis considers VN arrivals and departures over a period of time. This provides insight into the number of accepted VNs for different compared variants. We have developed an in-house discrete event simulator that generates simulation scenarios representing the arrival and departure of VNs. The VN arrival rate of each scenario follows a Poisson distribution with a given mean. We vary the mean of Poisson distribution between 4 – 12 VNs per 100 time units. VN life time in all the scenarios is exponentially distributed with a mean of 100 time units. The number of VNodes of each arrived VN is kept at 8, whereas, the LNR of the VN is chosen randomly between 1 and 3.5. The simulation time of each scenario is 10000 time units,

and we exclude measurements from the first 1000 time instances to capture the steady state performance. For each problem instance, we generate 5 random simulation scenarios and report mean performance metrics to gain statistical confidence. These parameters have been chosen in accordance with those used in the network virtualization literature.

6.1.4 Latency constraint generation

For each VN, we generate $|\bar{E}|$ VPath-based latency constraints. To do so, we sort all the shortest VPaths between all pairs of VNodes in non-increasing order (in terms of the number of VLinks) and choose the first $|\bar{E}|$ paths. For each selected VPath $\bar{a}$, we generate a latency budget $\ell^{(\bar{a})}$ by leveraging the set of input candidate SPaths for each VLink $\bar{e} \in \bar{a}$, which ensures the existence of a feasible latency budget for embedding. Otherwise, an arbitrarily generated latency budget can be too strict (or too relaxed) causing none (or all) of the candidate SPaths to satisfy that budget. If $p^*_{\bar{e}}$ is the SPath with shortest distance among the set of candidate SPaths for a VLink $\bar{e} \in \bar{a}$, then the latency budget for the VPath $\bar{a}$ is generated as follows:

$$\ell = (\bar{a}, L): \quad \ell^{(L)} = \alpha \sum_{\forall \bar{e} \in \bar{a}} L_{p^*_{\bar{e}}} \tag{6.1}$$

$1.0 \leq \alpha \leq 2.0$ is a tuning parameter allowing us to vary the strictness of the latency constraints. For instance, $\alpha = 1.0$ implies only the shortest candidate SPath $p^*_{\bar{e}}$ for each VLink in $\bar{a}$ can satisfy the latency constraints. When $\alpha > 1.0$, (6.1) increases the latency bound for a VPath $\bar{a}$ allowing more candidate SPaths for each VLink to satisfy the constraints.

Table 6.1: Compared Variants

Variant Name	**Latency Constraints**	**Differential Delay Constraints**
L(∞)-DD(∞)	$\alpha = \infty$	DD_{max}=∞
L(α)-DD(∞)	Variable α	DD_{max}=∞
L(∞)-DD(250)	$\alpha = \infty$	DD_{max}=250 μs
L(∞)-DD(10)	$\alpha = \infty$	DD_{max}=10 μs
L(α)-DD(250)	Variable α	DD_{max}=250 μs
L(α)-DD(10)	Variable α	DD_{max}=10 μs

6.1.5 Compared Variants

We consider six problem variants representing different combinations of latency and differential delay constraints (Table 6.1). L(∞)-DD(∞) is the baseline that considers neither latency nor differential delay constraints, whereas, L(α)-DD(∞) imposes only latency constraints with varying α. In contrast, L(∞)-DD(250) and L(∞)-DD(10) consider only differential delay constraints. Depending on the different values of DD_{max} imposed by the enabling technologies discussed in Section 3.4.3, there are two more variants L(∞)-DD(250) and L(∞)-DD(10). Finally, L(α)-DD(250) and L(α)-DD(10) apply both constraints for varying α with the corresponding values of DD_{max}, respectively.

6.2 Evaluation Metrics

Spectrum slice usage (SSU) The percentage of spectrum slices allocated to a VN embedding with respect to the total number of slices in the SN.

Avg. number of splits used for a VLink (NSU) The ratio of the total number of splits used to embed the VLinks of a VN to the total number of VLinks in a VN.

Avg. number of distinct SPaths used for a VLink (NDP) The ratio of the total number of distinct SPaths used to map the VLinks of a VN to the number of VLinks in a VN.

Execution Time The time required for an algorithm to find a VN embedding.

Blocking ratio The fraction of VN requests over all the VN requests that could not be embedded on the EON.

Cost ratio The ratio of costs obtained by two different approaches for solving the same problem instance, where cost is computed using (4.10).

6.3 Microbenchmarks

6.3.1 Impact of latency and differential delay constraints

Fig. 6.2(a) shows the impact of latency constraint (α=1.25) on SSU, compared to the case where VNs do not apply any latency guarantee (L(∞)-DD(∞)), for both fixed- and flex- grid EON. To show the actual impact, ILP formulation is executed on VNs with LNR ranging from 1 to 2.5. We observe that relatively-sparse VNs (*i.e.*, LNR $<$ 2) do not exhibit much increase in SSU for both fixed- and flex-grid, while denser VNs (*i.e.*, LNR $\geq$ 2) result in higher SSU when subject to latency guarantees. This is because VLinks in dense VNs cannot always use the most spectrally-efficient SPaths due to spectrum resource exhaustion/fragmentation on those SPaths. The increase in SSU is more marked for fixed-grid due to its lack of flexibility in transmission configuration and spectrum allocation. Finally, note that, in the solutions obtained with L(∞)-DD(∞), up to 32% of latency constraints would be violated, which demonstrates the importance of having an explicit enforcement of latency bounds.

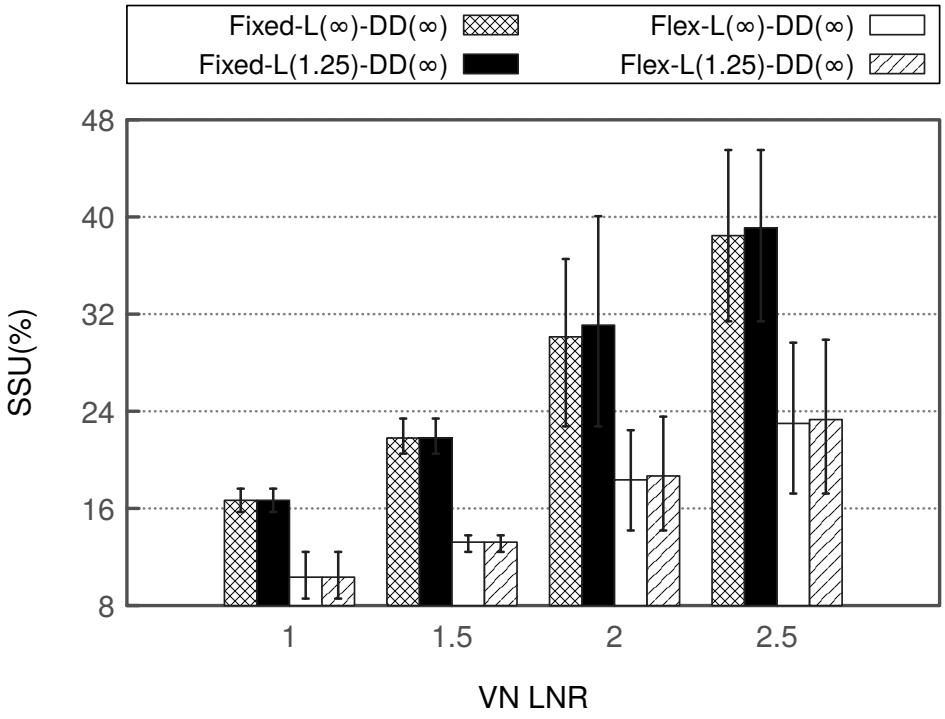
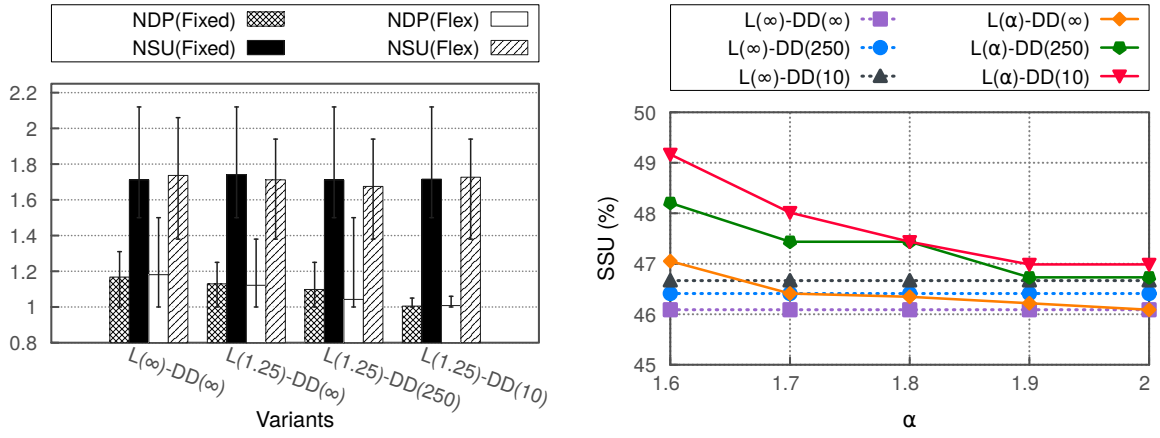


Figure 6.2: Impact of VN density

Fig. 6.3(a) shows how NDP and NSU are affected by the differential delay constraints. For instance, $DD_{max} = 10$ forces the splits of a VLink to use SPaths that differ by at most 2km in length. To ensure that in Nobel Germany SN, the same SPath is used by all the splits of a VLink in most cases of our simulation, as reflected by the values of NDP ($\approx$1) and NSU ($>$1) incurred by L(∞)-DD(10). Such strict requirement also rendered some of the VN embeddings infeasible. The NDP for L(∞)-DD(250) does not significantly differ from that of L(∞)-DD(∞). In fact, since the objective function minimizes spectrum usage, shorter SPaths are preferred whenever possible and splits with high length difference among candidate SPaths are selected rarely.

We observed that considering differential delay constraints does not significantly impact resource usage, *i.e.*, SSU. This is because the first few candidate SPaths (sorted in increasing order of length) of a VLink had sufficient capacity in most cases, allowing VLinks to be mapped on them. In order to force the VLinks to use a wider range of SPaths, we deliberately generated a set of VNs that maximize the VLinks with overlapping candidate SPaths, to increase the chances of having bottleneck SLinks along the first few candidate SPaths. For this scenario, Fig. 6.3(b) presents the SSU (obtained using the ILP) on fixed-grid while varying the latency budget. Even in such a scenario, we do not observe significant impact of the differential delay constraint on SSU. Rather, latency constraints have larger impact on SSU as the α governs which SPaths can be used in the embedding of a VLink.



(a) Impact of diff. delay

(b) Impact of latency and diff. delay

Figure 6.3: Small scale analysis

6.3.2 Comparison between Heuristic and Optimal solution

Our cost function (4.10) is dominated by resource usage in the EON. Therefore, cost ratio between heuristic and ILP gives an empirical measure of how much additional resources are allocated by the heuristic. We present the cumulative distribution function (CDF) of cost ratios in Fig. 6.4(a). Over all instances, the heuristic resulted in only 2.5% and 0.8% additional cost on average compared to the optimal solution for fixed- and flex-grid EONs, respectively.

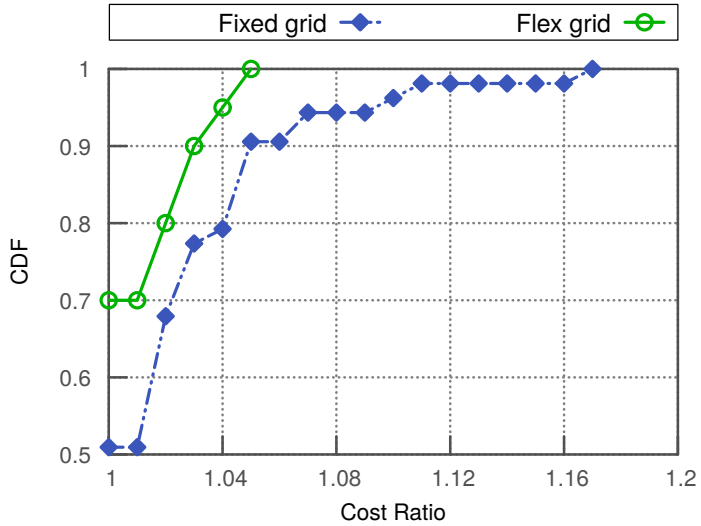
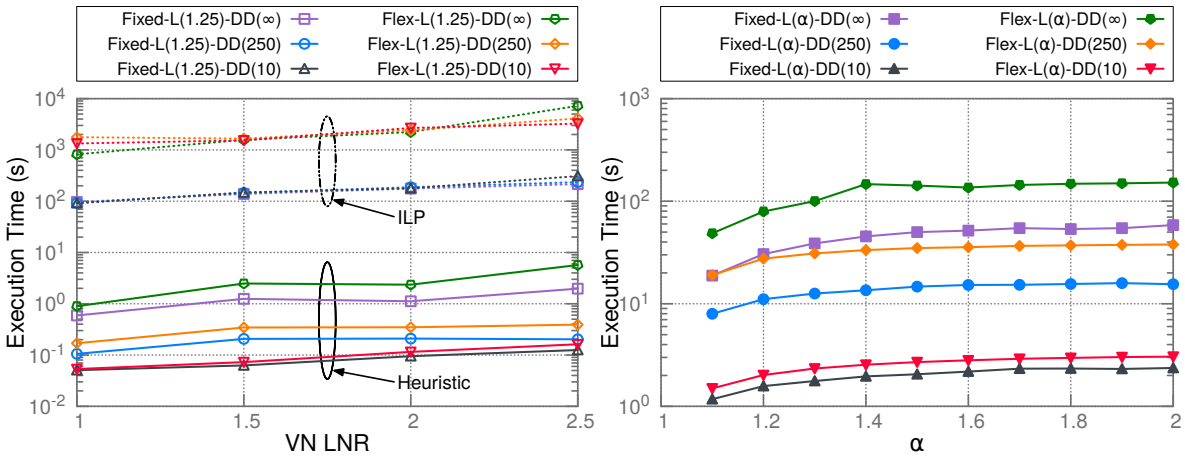


Figure 6.4: Cost ratio of the heuristic compared to ILP

6.3.3 Scalability Analysis

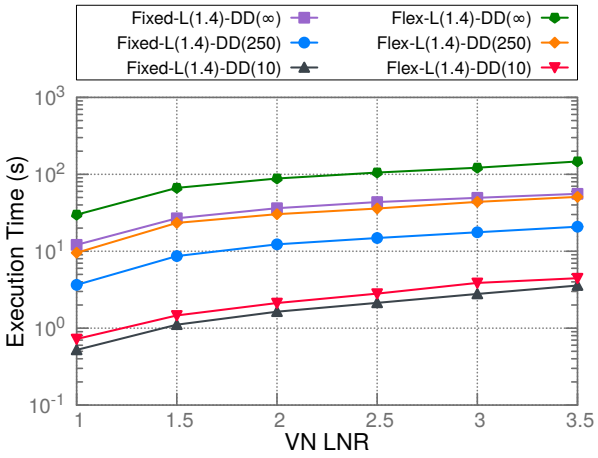
Fig. 6.5(a) presents a comparison between the heuristic's and the ILP's execution time for solving the same problem instances across different variants. In general, flex-grid increases complexity due to higher number of data rates and finer spectrum granularity, resulting in increased running time for both ILP and heuristic. Higher LNR implies a larger number of VLinks, hence, the larger execution time. ILP's execution time is not influenced by the adoption of differential delay constraint. However, the heuristic becomes faster for the same scenarios as it prunes the search space by considering only the combination of SPaths that satisfy differential delay constraint. In all cases, heuristic was observed to run 3 to 4 orders of magnitude faster than ILP. We also run the heuristic on much larger problem instances (described in Section 6.1.1) and report the

execution time by varying α in Fig. 6.5(b) and by varying VN LNR in Fig. 6.5(c). Fig. 6.5(b) shows that increasing α expands heuristic's search space by allowing more SPaths as the potential candidates, resulting in increased running time. Fig. 6.5(c) shows that even for a VN with as many as 175 VLinks (*i.e.*, VN LNR 3.5), the heuristic finds a solution in ≈2 minutes.



(a) Impact of VN density in small scale

(b) Impact of α in large scale

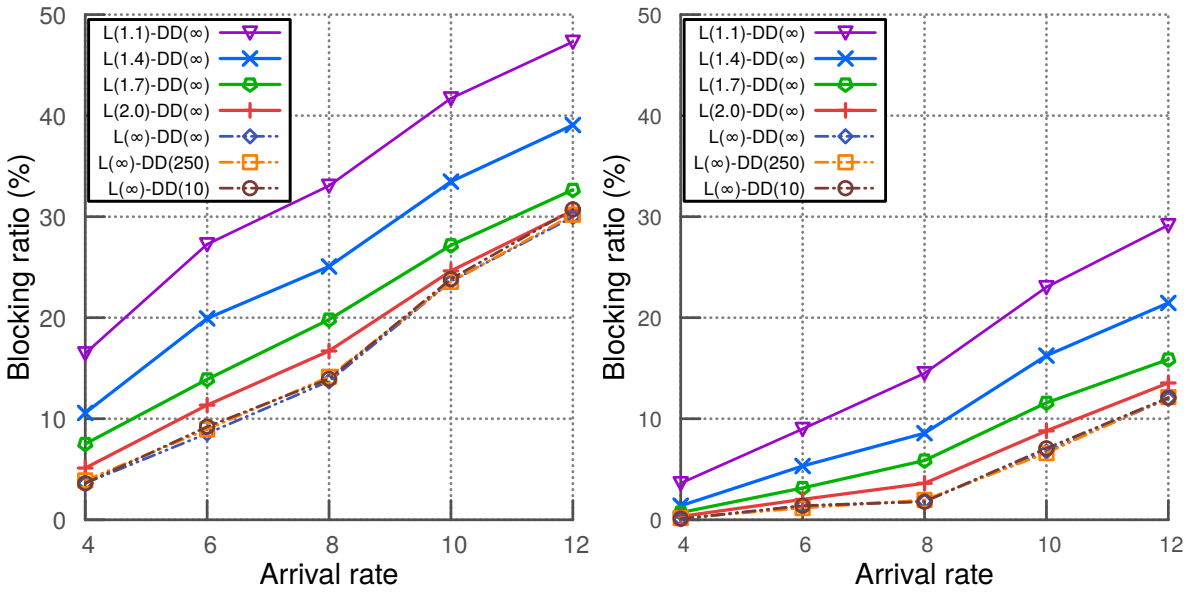


(c) Impact of VN density in large scale

Figure 6.5: Scalability analysis

6.4 Steady State Analysis

We simulate the arrival of VNs at different rates on both fixed- and flex-grid EON and embed a VN as it arrives using the heuristic. Fig. 6.6 presents the blocking ratio for a number of variants by varying arrival rate of VNs. L(∞)-DD(∞), our baseline variant, yields the lowest blocking ratio, but does not guarantee that all the latency constraints will be satisfied. Among the compared variants, the one with the strictest latency constraint (*i.e.*, $\alpha = 1.1$) results in the highest blocking ratio, *i.e.*, 20% and 12% more on average than the baseline for fixed- and flex-grid EONs. This is because only a small fraction of the candidate SPaths of a VLink can satisfy such latency constraints, which also skews the resource usage on the SLinks of those SPaths. Relaxing the latency constraint (*i.e.*, increasing α) allows the VNs to be embedded using other (potentially longer) SPaths avoiding the bottleneck SLinks, hence, blocking ratio decreases for higher α. Applying only differential delay constraints does not significantly impact the blocking ratio, which conforms with our previous finding on SSU. Unlike latency constraints, differential delay constraints do not reduce the size of the candidate SPath pool for a VLink, thus yields a blocking ratio similar to the baseline variant.

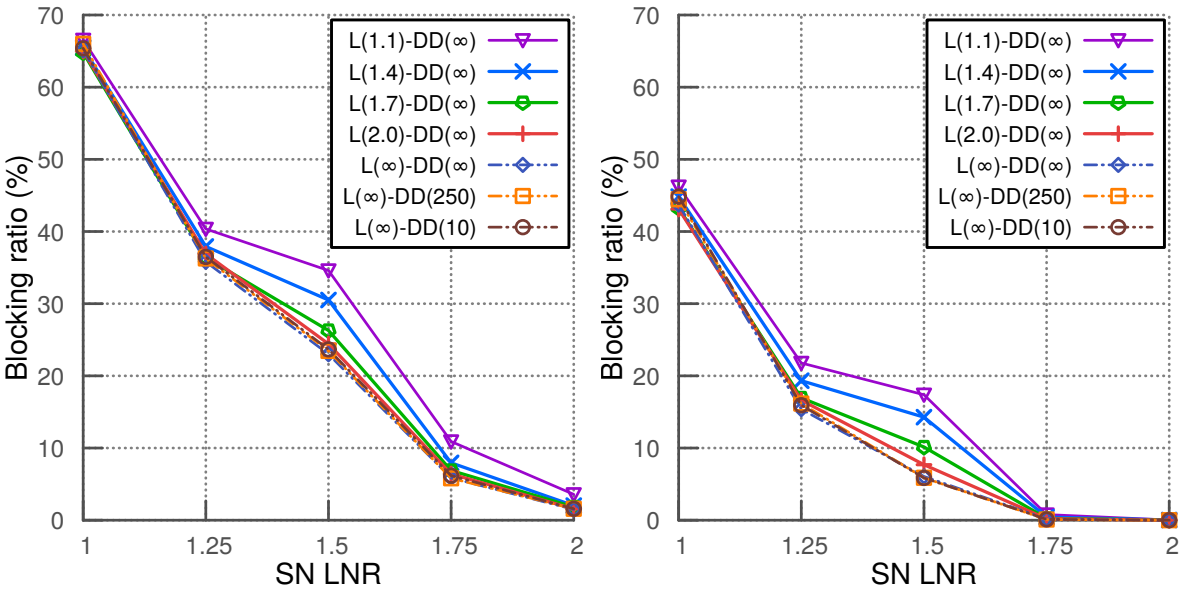


(a) Fixed-grid (b) Flex-grid

Figure 6.6: Impact of arrival rate and latency constraints

We also vary SN density (*i.e.*, SN LNR) for a fixed arrival rate of 10 and present the re-

sulting blocking ratio for different variants in Fig. 6.7. For a sparse SN, *i.e.*, a ring SN, only 2 SPaths are available for mapping each VLink and candidate SPaths of different VLinks overlap with each other. Consequently, spectrum resources in SLinks exhaust faster, forcing all the variants to block similar percentages of VNs. With the increase in SN LNR and consequently spectrum resources up to a certain point, spectrum capacity no longer remains a bottleneck and blocking ratio becomes dominated by latency constraints, yielding a similar behavior as the one in Fig. 6.6. Finally, as the SN become denser, higher SPath diversity eliminates the impact of latency constraints, resulting in almost 100% VN requests to be accepted.



(a) Fixed-grid (b) Flex-grid

Figure 6.7: Impact of SN density and latency constraints

Chapter 7

Conclusion and Future Work

7.1 Conclusion

Motivated by the increasing demand for low-latency services, in this thesis, we addressed the problem of VN embedding over EON with latency guarantees. We considered both fixed- and flex-grid EONs with flexible transmission parameters, namely baud rate, modulation format, and FEC overhead to efficiently allocate resources to VLinks. We presented a novel approach that constrains latency over virtual paths, instead of bounding the latency on each virtual links, to better capture application latency requirements. We also identified latency contributions of various active/passive elements in an EON. To analyze the impact of latency constraints on resource allocation, we presented an ILP formulation for embedding VNs that respect latency and differential delay constraints when using multiple SPaths to satisfy VLink demands. We also devised a heuristic solution to address the computational complexity of the ILP formulation. Our proposed heuristic performs close to the ILP formulation while executing several orders of magnitude faster. In static scenarios, our extensive simulations, based on realistic reaches, transmission configurations, and topologies, show that guaranteeing latency does not require significant additional resources. However, in a dynamic scenario, an additional 20% and 12% VN requests were blocked for fixed- and flex-grid EONs (Nobel Germany topology), respectively, when subject to latency constraints. We also showed that by adding 15% SLinks to the unmodified Nobel Germany topology, we could bring the blocking ratio down to 10% and 1% for fixed- and flex-grid EONs, respectively.

7.2 Future Work

There are several promising directions for future research as follows:

Variable VNode mapping: In this thesis, we considered fixed node mapping, i.e., each VNode is mapped to an SNode as part of the VN mapping requirement. Instead of a fixed node mapping each VNode could have multiple SNodes options to be mapped to. However, having a variable node mapping increases the complexity of the problem significantly.

Cost functions: In our work, we try to find a VN embedding which minimizes the resource allocation in terms of number of spectrum slices. This approach could lead to spectrum fragmentation; consequently, embedding other VNs could be more costly or even infeasible due to the contiguity and continuity constraints. One way to overcome this issue is by changing the objective function that considers the spectrum fragmentation as a part of the cost.

Admission control: For the dynamic embedding, we simply accept and embed any VN which could be embedded using our heuristic. One possible issue with this approach is that accepting a high demand VN could lead to rejecting many low demand VNs. This issue could be addressed by having an admission control which accepts VNs based on some criteria such as expected revenue to maximize the benefit in dynamic embedding.

Batch embedding: Our proposed solution only tackles the problem of embedding a single VN at a time. An interesting extension to our work is VNE for multiple VNs at the same time considering different cost functions, such as acceptance ratio, resource utilization, spectrum fragmentation and so on.

References

[1] Flex ethernet 2.0 implementation agreement [online] https://www.oiforum.com/wp-content/uploads/2019/01/oif-flexe-02.0-1.pdf.

[2] Sndlib repository [online] http://sndlib.zib.de/home.action.

[3] Information week – wall street's quest to process data at the speed of light, 2007.

[4] Norberto Amaya, Georgios Zervas, and Dimitra Simeonidou. Introducing node architecture flexibility for elastic optical networks. *IEEE/OSA Journal of Optical Communications and Networking*, 5(6):593–608, 2013.

[5] GSA Global Mobile Suppliers Association. The road to 5g: Drivers, applications, requirements and technical development. White paper, 2015.

[6] Sara Ayoubi, Chadi Assi, Khaled Shaban, and Lata Narayanan. Minted: Multicastvirtualnetworkembedding in cloud data centers withdelay constraints. *IEEE Transactions on Communications*, 63(4):1291–1305, 2015.

[7] Greg Bernstein, Diego Caviglia, Richard Rabbat, and Huub Van Helvoort. Vcat-lcas in a clamshell. *IEEE Communications Magazine*, 44(5):34–36, 2006.

[8] Francesco Bianchi and Francesco Lo Presti. A markov reward based resource-latency aware heuristic for the virtual network embedding problem. *ACM SIGMETRICS Performance Evaluation Review*, 44(4):57–68, 2017.

[9] Vjaceslavs Bobrovs, Sandis Spolitis, and Girts Ivanovs. Latency causes and reduction in optical metro networks. In *Optical Metro Networks and Short-Haul Systems VI*, volume 9008, page 90080C. International Society for Optics and Photonics, 2014.

[10] Raouf Boutaba, Nashid Shahriar, and Siavash Fathi. Elastic optical networking for 5G transport. *Springer Journal of Network and Systems Management*, 25(4):819–847, 2017.

[11] Richard A Brualdi. Introductory combinatorics. *New York*, 3, 1992.

[12] Bijoy Chand Chatterjee et al. Routing and spectrum allocation in elastic optical networks: A tutorial. *IEEE Comm. Surveys & Tutorials*, 17(3):1776–1800, 2015.

[13] Xiaomin Chen, Admela Jukan, and Ashwin Gumaste. Multipath de-fragmentation: Achieving better spectral efficiency in elastic optical path networks. In *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)*, pages 390–394, 2013.

[14] Giorgos Chochlidakis and Vasilis Friderikos. Low latency virtual network embedding for mobile networks. In *Proceedings of IEEE International Conference on Communications (ICC)*, pages 1–6, 2016.

[15] Giorgos Chochlidakis and Vasilis Friderikos. Mobility aware virtual network embedding. *IEEE Transactions on Mobile Computing*, 16(5):1343–1356, 2016.

[16] NM Mosharaf Kabir Chowdhury and Raouf Boutaba. A survey of network virtualization. *Computer Networks*, 54(5):862–876, 2010.

[17] NM Mosharaf Kabir Chowdhury, Muntasir Raihan Rahman, and Raouf Boutaba. Virtual network embedding with coordinated node and link mapping. In *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)*, pages 783–791, 2009.

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

[19] Xenofon Foukas, Georgios Patounas, Ahmed Elmokashfi, and Mahesh K Marina. Network slicing in 5g: Survey and challenges. *IEEE Communications Magazine*, 55(5):94–100, 2017.

[20] Ori Gerstel, Masahiko Jinno, Andrew Lord, and SJ Ben Yoo. Elastic optical networking: A new dawn for the optical layer? *IEEE Communications Magazine*, 50(2), 2012.

[21] Gwang-Hyun Gho, Lauren Klak, and Joseph M Kahn. Rate-adaptive coding for optical fiber transmission systems. *Journal of Lightwave Technology*, 29(2):222–233, 2010.

[22] Long Gong and Zuqing Zhu. Virtual optical network embedding (VONE) over elastic optical networks. *IEEE/OSA Journal of Lightwave Technology*, 32(3):450–460, 2014.

[23] Khaled Hejja and Xavier Hesselbach. Online power aware coordinated virtual network embedding with 5G delay constraint. *Elsevier Journal of Network and Computer Applications*, 124:121–136, 2018.

[24] Sheng Huang, Charles U Martel, and Biswanath Mukherjee. Survivable multipath provisioning with differential delay constraint in telecom mesh networks. *IEEE/ACM Transactions On Networking*, 19(3):657–669, 2011.

[25] Huawei. 5g: A technology vision. White paper.

[26] Infinera. Low latency – how low can you go? White paper, 2016.

[27] Johannes Inführ and Günther R Raidl. Introducing the virtual network mapping problem with delay, routing and location constraints. In *Network optimization*, pages 105–117. Springer, 2011.

[28] Karthikeswar Ivaturi and Tilman Wolf. Mapping of delay-sensitive virtual networks. In *Proceedings of IEEE International Conference on Computing, Networking and Communications (ICNC)*, pages 341–347, 2014.

[29] John Jay. Low signal latency in optical fiber networks. In *Corning Optical Fiber, Proc. of the 60th IWCS, Conference, Charlotte, NC, USA,(6-9 Nov. 2011)*. Citeseer, 2011.

[30] M. Jinno, B. Kozicki, H. Takara, A. Watanabe, Y. Sone, T. Tanaka, and A. Hirano. Distance-adaptive spectrum resource allocation in spectrum-sliced elastic optical path network [topics in optical communications]. *IEEE Communications Magazine*, 48(8):138–145, August 2010.

[31] Maria A Lema, Andres Laya, Toktam Mahmoodi, Maria Cuevas, Joachim Sachs, Jan Markendahl, and Mischa Dohler. Business case and technology analysis for 5G low latency applications. *IEEE Access*, 5:5917–5935, 2017.

[32] Rongping Lin, Shan Luo, Jingwei Zhou, Sheng Wang, Anliang Cai, Wen-De Zhong, and Moshe Zukerman. Virtual network embedding with adaptive modulation in flexi-grid networks. *IEEE/OSA Journal of Lightwave Technology*, 36(17):3551–3563, 2017.

[33] Víctor López and Luis Velasco. *Elastic Optical Networks*. Architectures, Technologies, and Control, Switzerland: Springer Int. Publishing, 2016.

[34] Waseem Mandarawi, Andreas Fischer, Amine Mohamed Houyou, Hans-Peter Huth, and Hermann De Meer. Constraint-based virtualization of industrial networks. In *Principles of Performance and Reliability Modeling and Evaluation*, pages 567–586. Springer, 2016.

[35] Nokia. 5g use cases and requirements. White paper.

[36] Optelian. A sensible low-latency strategy for optical transport networks. White paper, 2014.

[37] Albert Pagès, Jordi Perelló, Salvatore Spadaro, and Jaume Comellas. Optimal route, spectrum, and modulation level assignment in split-spectrum-enabled dynamic elastic optical networks. *IEEE/OSA Journal of Optical Communications and Networking*, 6(2):114–126, 2014.

[38] Imtiaz Parvez, Ali Rahmati, Ismail Guvenc, Arif I Sarwat, and Huaiyu Dai. A survey on low latency towards 5g: Ran, core network and caching solutions. *IEEE Communications Surveys & Tutorials*, 20(4):3098–3130, 2018.

[39] Sriram Pemmaraju and Steven Skiena. *Computational Discrete Mathematics: Combinatorics and Graph Theory with Mathematica®*. Cambridge university press, 2003.

[40] Y. Pointurier, N. Benzaoui, W. Lautenschlaeger, and L. Dembeck. End-to-end time-sensitive optical networking: Challenges and solutions. *IEEE/OSA Journal of Lightwave Technology*, 37(7):1732–1741, April 2019.

[41] Viraf Reporter. The value of a millisecond: Finding the optimal speed of a trading infrastructure. Vision note, April 2008.

[42] Cristina Rottondi, Pierpaolo Boffi, Paolo Martelli, and Massimo Tornatore. Routing, modulation format, baud rate and spectrum allocation in optical metro rings with flexible grid and few-mode transmission. *IEEE/OSA Journal of Lightwave Technology*, 35(1):61–70, 2017.

[43] N Sambo, G Meloni, F Cugini, F Fresi, A D'Errico, L Poti, P Iovanna, and P Castoldi. Routing, code, and spectrum assignment, subcarrier spacing, and filter configuration in elastic optical networks. *IEEE/OSA Journal of Optical Communications and Networking*, 7(11):B93–B100, 2015.

[44] Nicola Sambo et al. Routing code and spectrum assignment (RCSA) in elastic optical networks. *Journal of Lightwave Technology*, 33(24):5114–5121, 2015.

[45] Nashid Shahriar, Sephr Taeb, Shihabur Rahman Chowdhury, Massimo Tornatore, Raouf Boutaba, Jeebak Mitra, and Mahdi Hemmati. Achieving a Fully-Flexible Virtual Network Embedding in Elastic Optical Networks. In *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)*, 2019.

[46] Liao Shengquan, Wu Chunming, Zhang Min, and Jiang Ming. An efficient virtual network embedding algorithm with delay constraints. In *Proceedings of IEEE International Symposium on Wireless Personal Multimedia Communications (WPMC)*, pages 1–6, 2013.

[47] Takahito Tanimura, Takeshi Hoshida, Tomoyuki Kato, Shigeki Watanabe, Makoto Suzuki, and Hiroyuki Morikawa. Throughput and latency programmable optical transceiver by using dsp and fec control. *OSA Optics express*, 25(10):10815–10827, 2017.

[48] Akamai Technologies. State of online retail performance – 2017 holiday retrospective. White paper, 2017.

[49] Yang Wang, Xiaojun Cao, and Yi Pan. A study of the routing and spectrum allocation in spectrum-sliced elastic optical path networks. In *INFOCOM, 2011 Proceedings IEEE*, pages 1503–1511. IEEE, 2011.

[50] Tianhua Xu, Gunnar Jacobsen, Sergei Popov, Jie Li, Evgeny Vanin, Ke Wang, Ari T Friberg, and Yimo Zhang. Chromatic dispersion compensation in coherent transmission system using digital filters. *OSA Optics express*, 18(15):16243–16257, 2010.

[51] Guoying Zhang, Marc De Leenheer, Annalisa Morea, and Biswanath Mukherjee. A survey on ofdm-based elastic core optical networking. *IEEE Communications Surveys & Tutorials*, 15(1):65–87, 2012.

[52] Min Zhang, Chunming Wu, Ming Jiang, and Qiang Yang. Mapping multicast service-oriented virtual networks with delay and delay variation constraints. In *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM)*, pages 1–5, 2010.

[53] Shuqiang Zhang, Rui Wang, Uttam Mandal, M Farhan Habib, and Biswanath Mukherjee. Connecting the clouds with low-latency, low-cost virtual private lines enabled by sliceable optical networks. In *Proceedings of IEEE Global Communications Conference (GLOBECOM)*, pages 2370–2375, 2013.

[54] Juzi Zhao, Suresh Subramaniam, and Maïté Brandt-Pearce. Virtual topology mapping in elastic optical networks. In *Proceedings of IEEE International Conference on Communications (ICC)*, pages 3904–3908, 2013.