Universidad Politécnica de Cartagena

Department of Information and Communication Technologies

Novel SDN and NFV Techniques for Optical Network Orchestration

Ph.D. Thesis

José Juan Pedreño Manresa

Advisors

Prof. Pablo Pavón Mariño

Dr. José Luis Izquierdo Zaragoza

2019

**CONFORMIDAD DE SOLICITUD DE AUTORIZACIÓN DE DEPÓSITO DE
TESIS DOCTORAL POR EL/LA DIRECTOR/A DE LA TESIS**

D. Pablo Pavón Mariño, y D. José Luis Izquierdo Zaragoza, Directores de la Tesis doctoral “Novel SDN and NFV Techniques for Optical Network Orchestration”.

INFORMA:

Que la referida Tesis Doctoral, ha sido realizada por D. José Juan Pedreño Manresa, dentro del Programa de Doctorado Tecnologías de la Información y las Comunicaciones, dando mi conformidad para que sea presentada ante el Comité de Dirección de la Escuela Internacional de Doctorado para ser autorizado su depósito.

☒ Informe positivo sobre el plan de investigación y documento de actividades del doctorando/a emitido por el Director/ Tutor (**RAPI**).

La rama de conocimiento en la que esta tesis ha sido desarrollada es:

- ☐ Ciencias
- ☐ Ciencias Sociales y Jurídicas
- ☒ Ingeniería y Arquitectura

En Cartagena, a 11 de julio de 2019

EL DIRECTOR DE LA TESIS

Fdo: Pablo Pavón Mariño

EL CODIRECTOR DE LA TESIS

Fdo: José Luis Izquierdo Zaragoza

COMITÉ DE DIRECCIÓN ESCUELA INTERNACIONAL DE DOCTORADO

CONFORMIDAD DE DEPÓSITO DE TESIS DOCTORAL
POR LA COMISIÓN ACADÉMICA DEL PROGRAMA

D. **Jorge Larrey Ruiz**, Presidente/a de la Comisión Académica del Programa de Doctorado en Tecnologías de la Información y las Comunicaciones.

INFORMA:

Que la Tesis Doctoral titulada, “**Novel SDN and NFV Techniques for Optical Network Orchestration**”, ha sido realizada, dentro del mencionado Programa de Doctorado, por D/D^a **José Juan Pedreño Manresa**, bajo la dirección y supervisión del **Dr. Pablo Pavón Mariño** y el **Dr. José Luis Izquierdo Zaragoza**.

En reunión de la Comisión Académica, visto que en la misma se acreditan los indicios de calidad correspondientes y la autorización del Director/a de la misma, se acordó dar la conformidad, con la finalidad de que sea autorizado su depósito por el Comité de Dirección de la Escuela Internacional de Doctorado.

☒ Evaluación positiva del plan de investigación y documento de actividades por el Presidente de la Comisión Académica del programa (**RAPI**).

La Rama de conocimiento por la que esta tesis ha sido desarrollada es:

- ☐ Ciencias
- ☐ Ciencias Sociales y Jurídicas
- ☒ Ingeniería y Arquitectura

En Cartagena, a 9 de julio de 2019

EL PRESIDENTE DE LA COMISIÓN ACADÉMICA

**JORGE|
LARREY|
RUIZ**

Firmado digitalmente por JORGE|
LARREY|RUIZ
Nombre de reconocimiento (DN):
cn=JORGE|LARREY|RUIZ
serialNumber=
givenName=JORGE, sn=LARREY RUIZ,
ou=Ciudadanos, o=ACCV, c=ES
Fecha: 2019.07.09 11:47:22 +02'00'

Fdo: Jorge Larrey Ruiz

COMITÉ DE DIRECCIÓN ESCUELA INTERNACIONAL DE DOCTORADO

In this country, you gotta make the money first. Then when you get the money, you get the power. Then when you get the power, then you get the women.

Tony Montana

People often take for granted their high-speed Internet connections, but they would be horrified if they found out that underneath, everything is the digital equivalent of duct tape and plastic ties.

Anonymous

Abstract

Multilayer IP-over-WDM networks are the foundation of today’s metro and core networks, and the only enabling technology for the bandwidth explosion in the 5G mobile backhaul. The planning and operation of these networks have been traditionally based on separate IP and optical approaches, largely relying on capacity over-provisioning and manual operation processes. This methodology is no longer valid for the stringent requirements in terms of bandwidth, end-to-end latency, dynamicity, scale and operation complexity of 5G networks and beyond. On the contrary, a mix of evolution and revolution is required in IP-over-WDM planning and operation principles to meet these challenges.

This Ph.D. thesis contributes to this field with *Novel SDN and NFV Techniques for Optical Network Orchestration*. We focus on network planning and operation of access, metro, and backbone multilayer networks, including case studies where traditional network resources are allocated along Information Technology (IT) resources to guarantee good Quality of Service (QoS). We also explore the importance of network automation, efficient orchestration, and agile operation as a basis for sustainable network growth.

Particularly, this work covers a four-year research period on several relevant topics on multilayer optical network orchestration: (i) Network Planning, (ii) Multilayer Dynamic Operation Using Software-Defined Networking and (iii) Dynamic VNF Placement and Operation on 5G Metro-Access Networks.

First, we analyze Network Planning, discussing how multilayer networks can be planned efficiently to withstand the effect of network failures. With the emergence of the aforementioned challenges, network planning goes beyond “adding more capacity”. Contributions in this part have the objective to properly plan and dimension networks that guarantee not only survivability but also end-to-end latency and QoS. Both transport (see Chapter 3) and metropolitan area networks (see Chapters 4 and 5) are covered in our analyses, tailoring our solutions to the specific requirements and characteristics of each network segment.

Second, we explore Multilayer Dynamic Operation Using Software-Defined Networking. This new paradigm is a game-changer in terms of network management and operation. Network orchestration is studied using two different approaches in the form of proofs-of-concept (PoCs), leveraging Net2Plan extensions. First, the orchestration of OpenFlow-based networks is achieved by creating an off-platform application on top of the OpenDaylight controller. Then, we experiment with dynamic multilayer network op-

eration using an active stateful multilayer Path Computation Element (PCE), based on protocols such as Border Gateway Protocol with Link-State extensions (BGP-LS) and Path Computation Element Protocol (PCEP), integrated into an Application-Based Network Operations (ABNO) architecture.

Lastly, we discuss Dynamic VNF Placement and Operation on 5G Metro-Access Networks. Dynamic scenarios with high mobility of users and rapid changes in service expectations (e.g., football matches, concerts, massive parades) require a special and careful operation, especially taking into account the requirements expected by 5G. We present the idea of a joint orchestration of radio and cloud resources, exploring the benefits of efficient and dynamic virtual network function (VNF) deployment with the goal of satisfying QoS/quality of experience (QoE) requirements. Also, a novel VNF-handover mechanism is presented, showing the benefits of proactive VNF allocation to avoid service disruption.

Summarizing, this Ph.D. thesis opens the door to the so-called Software-Defined Network Function Virtualization (SDNFV). On the one hand, Part II and Part III demonstrate the capabilities to create *in-operation network planning* solutions to achieve the maximum performance of networks during their lifetime, enabled by proper network dimensioning and tight integration between planning tools and software-defined networking (SDN) controllers, as reported throughout this document. On the other hand, Part IV presents a new framework for operating a 5G-enabled access, where computing capabilities close to the customer edge are a must, and the demonstrated NFV handover is prominent. As a next step, putting together these two paradigms will enable end-to-end service orchestration, maximizing QoS/QoE whereas optimizing costs and resource utilization.

List of Contents

Abstract	i
List of Figures	vii
List of Tables	ix
List of Acronyms	xi
I Presentation	1
1 Introduction	3
1.1 Towards a New Era of Network Orchestration	3
1.2 Scope and Goals	4
1.3 Work Methodology	5
1.4 Outline	5
2 General Context	7
2.1 Technological Background	7
2.1.1 Multilayer Optical Networks	7
2.1.2 Software-Defined Networking	10
2.1.3 Network Function Virtualization	13
2.2 Evaluation Methodology	15
2.2.1 Network Planning	16
2.2.2 Network Simulation	16
2.2.3 Net2Plan	17

II	Network Planning	21
3	Traffic Survivability and Latency Awareness in Multilayer Network Design	23
3.1	Network Planning: When Survivability is not Enough	23
3.2	Case Study	25
3.2.1	Algorithm Design	25
3.2.2	IP-over-WDM Module and Recovery Schemes	28
3.2.3	Latency Computation	29
3.2.4	Testing Scenario	29
3.3	Conclusions and Further Work	33
4	Spectrum Management in Filterless Metro Networks	35
4.1	Filterless Optical Networks	35
4.1.1	Transponder Technologies	37
4.1.2	Traffic Dynamicity	38
4.2	Case Study	39
4.2.1	Algorithm Implementation	40
4.2.2	Spectrum Allocation Approaches	41
4.2.3	Testing Scenario	42
4.3	Conclusions and Further Work	44
5	Scalability of WaveBand-based ROADMs	47
5.1	Photonic Integrated WaveBand ROADMs	47
5.2	WB-ROADM Design	48
5.3	Case Study	49
5.3.1	ILP Formulation	50
5.4	Conclusions and Further Work	51
III	Multilayer Dynamic Operation Using Software-Defined Networking	53
6	Network Optimization using SDN Controllers	55
6.1	Off-platform Application for OpenDaylight	55
6.2	Communicating Net2Plan with OpenDaylight	57
6.3	Case Study	58
6.3.1	Retrieving Topology Information	59
6.3.2	Estimation of End-to-End Traffic Volumes	60
6.3.3	Real-Time Link Utilization Monitoring	61

List of Contents

6.3.4	Installing Forwarding Rules	61
6.3.5	Work Tracing	62
6.4	OpenDaylight and SDON	63
6.5	Conclusions and Further Work	64
7	Dynamic Operation of Multilayer Optical Networks	65
7.1	End-user Service Deployment by Network Operators	65
7.2	Application-Based Network Operations	66
7.3	Communicating Net2Plan with ABNO	68
7.4	Multilayer Dynamic Operation	70
7.5	Case Study	71
7.5.1	Workflow 1: Establishing PCC-PCE connection	71
7.5.2	Workflow 2: Service provisioning	71
7.5.3	Workflow 3: Reaction to fiber link failure	73
7.6	Conclusions and Further Work	73
IV	Dynamic VNF Placement and Operation on 5G Metro- Access Networks	75
8	Service Chaining in Mobility 5G Scenarios	77
8.1	The Dawn of the 5G Era	77
8.2	Joint Radio-Cloud Architecture	78
8.3	Joint Orchestration Case Study	80
8.3.1	Non-orchestrated Scenario	82
8.3.2	Orchestrated Scenario	85
8.4	NFV Handover Case Study	86
8.5	Conclusions and Further Work	88
V	Closing Discussion	91
9	Final Remarks	93
9.1	Main Contributions	93
9.2	Open Research Areas	94
9.2.1	SDNFV and End-to-end Service Orchestration	94
9.2.2	Delving in 5G Joint Operation	94
9.3	List of Publications	95
9.3.1	Publications in International Journals	95
9.3.2	Publications in International Conferences	96

References	99
-------------------	-----------

Appendices	111
-------------------	------------

A Reference Scenarios	113
------------------------------	------------

A.1 7-Node Spanish Network (Top7Spain)	113
--	-----

A.2 9-Node Internet2 Network (Internet2)	114
--	-----

A.3 14-Node National Science Foundation Network (NSFNet)	115
--	-----

A.4 15-Node Atlanta Network (Atlanta)	116
---	-----

List of Figures

1.1	Methodology	5
2.1	Network hierarchy classification	8
2.2	Example of a two-layered network	9
2.3	Spectrum fragmentation in optical networks	9
2.4	Planes of functionality in networking	11
2.5	Typical SDN architecture	12
2.6	OpenFlow Flow table and typical SDN architecture	14
2.7	Network function virtualization architecture	15
2.8	Offline network design & online simulation tool	20
3.1	Algorithm modules hierarchy	26
3.1	Pseudo-code for the algorithm construction phase	27
3.2	Pseudo-Code for the Local Search Phase	28
3.2	Total number of transponders for each topology (no latency limit versus a maximum limit of 62.5 ms)	32
3.3	Worst-case end-to-end latency distribution for each topology (no latency limit versus a maximum limit of 62.5 ms).	33
4.1	2-degree ROADM with a broadcast-and-select architecture	36
4.2	Full-filterless 2-degree node architecture	36
4.3	Resource allocation with different transponder technologies	38
4.4	Resource allocation at different time of the day/week	39
4.1	Pseudo-code for the Multi-Hour Spectrum Allocation algorithm	41
4.5	Reference horseshoe topology	42
4.6	Maximum throughput (in Tbps)	43
4.7	Spectrum occupation for four example demands	45
5.1	2-degree WB-ROADM	48

5.2	Reference architecture of “WB-Drop” blocks.	48
5.3	Filter performance (500 GHz pass-band) within the whole C band	49
5.4	8-node horseshoe topology.	51
5.5	Maximum throughput achieved (in Tbps)	52
6.1	OpenDaylight architecture	56
6.2	Architecture overview of the Net2Plan – OpenDaylight interconnection . .	57
6.3	OpenDaylight WebUI	59
6.4	Connection manager panel	60
6.5	Emulated topology	61
6.6	Emulated topology retrieved and shown in Net2Plan	62
6.7	Traces showing the message exchange for OpenDaylight	63
6.8	TransportPCE architecture	64
7.1	Workflow for deployment of end-user services	66
7.2	ABNO architecture	68
7.3	Architecture of the Net2Plan - ABNO interconnection	69
7.4	Screenshot of the implemented modules	70
7.5	Traces from Wireshark for the ABNO - Net2Plan experiment	72
7.6	Lightpath rerouting upon fiber failure	74
8.1	Multi-Tier Cloud-RAN Architecture	79
8.2	Feedback QoS loop	80
8.3	Distribution of micro-cells throughout the stadium seating area	81
8.4	Macro-cell distribution around the Camp Nou Stadium (Barcelona)	82
8.1	Pseudo-code for service request allocation and Service Chain deployment .	83
8.5	Hardware utilization of the non-orchestrated scenario	84
8.6	Blocking in each instant of the simulation	85
8.7	Hardware utilization of the orchestrated scenario	86
8.8	Average link utilization in both scenarios	87
8.9	Service chain handover problem	88
8.10	Comparison of average hardware utilization in NFV handover scenario . . .	89
A.1	7-node Spanish network (Spain7)	114
A.2	9-node Internet2 network (Internet2)	115
A.3	14-node National Science Foundation network (NSFNet)	116
A.4	15-node Atlanta network (Atlanta)	117

List of Tables

1.1	Goals and related sections	5
3.1	Overall throughput on NSFNet (in Tbps)	30
3.2	Overall throughput on Internet2 (in Tbps)	31
3.3	Overall throughput on Atlanta (in Tbps)	31
3.4	Proportion of IP demands and offered traffic violating latency constraints on NSFNet (latency unaware)	34
3.5	Proportion of IP demands and offered traffic violating latency constraints on Internet2 (latency unaware)	34
3.6	Proportion of IP demands and offered traffic violating latency constraints on Atlanta (latency unaware)	34
4.1	Throughput variation (percentage) when using semi-elastic allocation com- pared to other strategies	43
4.2	Throughput gain (percentage) when using dual resonator transponders in- stead of single resonator	44
8.1	Service type bandwidth requirement and associated service chain	81
8.2	Number of concurrent users per instance and hardware requirements per VNF	82
9.1	Publications in international journals	95
9.2	Publications in international conferences	96
A.1	Basic network information (Spain7)	114
A.2	Traffic matrix and link lengths in km (Top7Spain)	114
A.3	Basic network information (Internet2)	115
A.4	Traffic matrix and link lengths in km (Internet2)	115
A.5	Basic network information (NSFNet)	116
A.6	Traffic matrix (NSFNet)	116

A.7	Link lengths in km (NSFNet)	117
A.8	Basic network information (Atlanta)	118
A.9	Traffic matrix (Atlanta)	118
A.10	Link lengths in km (Atlanta)	118

List of Acronyms

ABNO	Application-Based Network Operations
AI	Artificial Intelligence
AMEN	Access-Metro Edge Node
API	Application Programming Interface
AR	Augmented Reality
ASE	Amplified Spontaneous Emission
ALTO	Application-Layer Traffic Optimization
BGP	Border Gateway Protocol
BGP-LS	Border Gateway Protocol with Link-State extensions
BRKGA	Biased Random-Key Genetic Algorithm
BVT	Bandwidth-Variable Transponder
CAPEX	CAPital EXpenditure
CDN	Content Delivery Network
CF	Central Frequency
CLI	Command-Line Interface
CMOS	Complementary Metal-Oxide-Semiconductor
CO	Central Office
CNIT	Consorzio Nazionale Interuniversitario per le Telecomunicazioni

DES	Discrete Event Simulation
DW	Drop and Waste
ECMP	Equal-Cost Multi-Path
EPC	Evolved Packet Core
ETSI	European Telecommunications Standards Institute
ERO	Explicit Route Object
FCAPS	Fault, Configuration, Accounting, Performance and Security
FON	Filterless Optical Network
FSR	Free Spectral Range
GA	Genetic Algorithm
GIRTEL	Telecommunication Networks Engineering Group
GRASP	Greedy Randomized Adaptive Search Procedure
GUI	Graphical User Interface
HTTP	HyperText Transfer Protocol
ILP	Integer Linear Programming
IoT	Internet of Things
IP	Internet Protocol
IPFRR	IP Fast ReRoute
IT	Information Technology
I2RS	Interface to the Routing System
JVM	Java Virtual Machine
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
LAN	Local Area Network
LCoS	Liquid Crystal on Silicon

List of Acronyms

LLDP	Link-Layer Discovery Protocol
LSP	Label-Switched Path
MANO	Management and Orchestration
MCEN	Metro-Core Edge Node
MEC	Mobile Edge Computing
MH-SA	Multi-Hour Spectrum Allocation
MILP	Mixed Integer Linear Programming
ML	Machine Learning
MPLS	Multi-Protocol Label Switching
MPLS-TE	Multi-Protocol Label Switching with Traffic Engineering extensions
MZI	Mach-Zehnder interferometer
MRR	Micro-Ring Resonators
NAT	Network Address Table
NBI	North Bound Interface
NF	Network Function
NFV	Network Function Virtualization
NMS	Network Managment System
NLRI	Network Layer Reachability Information
NOS	Network Operating System
NFVI	Network Function Virtualization Infrastructure
NF	Network Function
NETCONF	Network Configuration Protocol
OAM	Operations, Administration, and Management
OPEX	OPerational EXpenditure
OSPF	Open Shortest Path First

OTN	Optical Transport Network
OTT	Over-The-Top
OSS	Operational Support System
PCC	Path Computation Client
PCE	Path Computation Element
PCEP	Path Computation Element Protocol
PIC	Photonic Integrated Circuit
PoC	Proof-of-Concept
QoE	Quality of Experience
QoS	Quality of Service
RAN	Radio Access Network
REST	REpresentational State Transfer
RFC	Request For Comments
ROADM	Reconfigurable Optical Add/Drop Multiplexer
RSA	Routing and Spectrum Assignment
RWA	Routing and Wavelength Assignment
RX	Receiver
SA	Spectrum Allocation
SBI	South Bound Interface
SC	Service Chain
SDN	Software-Defined Networking
SDNFV	Software-Defined Network Function Virtualization
SDON	Software-Defined Optical Networking
SLA	Service-Level Agreement
SNMP	Simple Network Management Protocol

List of Acronyms

SOI	Silicon-On-Insulator
SRG	Shared-Risk Group
SFC	Service Function Chain
SF	Service Function
TBW	Transition Bandwidth
TE	Traffic Engineering
TED	Traffic Engineering Database
TRSA	Topology, Routing and Spectrum Assignment
TX	Transmitter
TXP	Transponder
UI	User Interface
UMTS	Universal Mobile Telecommunications System
UPCT	Universidad Politécnica de Cartagena
VNF	Virtual Network Function
VNF	Virtual Network Function
VNTM	Virtual Network Topology Manager
VM	Virtual Machines
WAN	Wide Area Network
WB-ROADM	WaveBand ROADM
WDM	Wavelength Division Multiplexing
WSS	Wavelength Selective Switch
XML	eXtensible Markup Language
YANG	Yet Another Next Generation

Part I

Presentation

Introduction

1.1	Towards a New Era of Network Orchestration	3
1.2	Scope and Goals	4
1.3	Work Methodology	5
1.4	Outline	5

1.1 Towards a New Era of Network Orchestration

The architecture of backbone networks has undergone a spectacular evolution in order to confront the different technological challenges that have emerged in the last 20 years. For a long time, scalability has been the main disadvantage faced by network operators. Considering that interannual traffic growth is over 20% [1] operators have mainly focused on deploying huge network infrastructures to support that traffic. The current reference model used by operators is based on multilayer designs, where the electronic equipment (IP/MPLS) that provides service to customers, is interconnected by end-to-end optical connections (or *lightpaths*) over an optical infrastructure.

Regarding the complexity and diversity of management tasks associated with these networks, traditionally, separate departments were in charge of each layer, with little to no communication between them [2]. Nonetheless, network operators have begun a migration process towards multilayer solutions, not only to optimize workflows for connection and provisioning services (which currently can take up to several weeks) but also to free themselves from proprietary solutions that hamper interoperability.

Software-Defined Networking (SDN) has emerged over the past few years as a key element for operators to improve their services and reduce costs. The success of the SDN paradigm is due to the industry's support of open source initiatives, and the emergence of controllers, centralized platforms capable of managing a multitude of network equipment by abstracting their control intelligence.

Unfortunately, SDN is far from being considered a mature technology. In its cur-

rent state, SDN has made partial progress in replacing proprietary command-line interfaces (CLIs) with a unified management environment [3, 4, 5]. In this context, advances have focused on the development of products, protocols, and languages to enable communication with controllers, even in a multi-vendor environment. However, at the beginning of this Ph.D. thesis (2014), very few initiatives had developed applications to provide controllers with its own intelligence, to meet the need of network operators. In other words, once there is the possibility of automating certain tasks, and the door is opened to semi-automatic optimization of network resources... How should we do this?

The general objective of this thesis is the development of novel techniques capable of orchestrating and optimizing multilayer optical networks, by taking advantage of SDN and NFV paradigms.

1.2 Scope and Goals

This Ph.D. thesis continued, in part, the work done by Prof. Pablo Pavón Mariño, who began working in the early 2000s in the field of multi-layer optical networking within the Telecommunication Networks Engineering Group (GIRTEL) [6] at Universidad Politécnica de Cartagena (UPCT) [7]. Moreover, this work can be considered a direct continuation of the one from Dr. José Luis Izquierdo Zaragoza, former student of Prof. Pavón and also co-advisor of this thesis. Henceforth, some preliminary goals for this Ph.D. were selected among the follow-ups of Dr. Izquierdo. Specifically, the following goals were chosen:

G1 **Design of multilayer orchestration algorithms**

This goal focuses on the design, implementation, and evaluation of algorithms capable of carrying out multiple daily tasks of a network operator; namely: the dynamic establishment of connections, reaction to faults, rerouting in case of congestion and so on. All of this in a multi-layer scenario, which adds a level of complexity due to the need to coordinate operation between layers.

G2 **Development of a network orchestrator using Net2Plan as a baseline**

The objective of this goal is to be able to leverage the capabilities of Net2Plan, originally developed as a network planning/simulation and simulation tool, to orchestrate multi-layer optical networks by using the software-defined networking paradigm.

G3 **Service chaining in dynamic scenarios**

This goal is related to the study of the challenges posed by dynamic scenarios, where the high mobility of users and rapid change of service requests make a static deployment of virtual network functions unsuitable. New techniques for NFV orchestration will be explored.

Starting from these goals, new and/or specific ones appeared as a result of research projects, collaboration, and other activities inside GIRTEL, and the scope slightly varied in order to cover other topics like filterless metro networks or devices like WaveBand ROADMs. Table 1.1 shows the sections where readers can be the contributions related to each one of the initial goals.

1.3. Work Methodology

Table 1.1: Goals and related sections

Goal	Reference to results
G1	Part II
G2	Part III
G3	Part IV

1.3 Work Methodology

Figure 1.1 depicts the methodology used during the development of this Ph.D. thesis, showing all steps from the conception of an idea, to the publication of results.

First, an idea is conceived; this idea is then refined and polished so the scope covers a specific goal. The next step is to define the metrics used to measure the problem, the scenario, and its outcome. Next steps involve the design of the solution and its implementation. The solution could be an algorithm or a specific piece of software. In this Ph.D. thesis, the Net2Plan [8] tool was widely used. This tool was developed as part of the Ph.D. thesis of Dr. José Luis Izquierdo Zaragoza (co-advisor of this work) and was used mainly used for network planning, simulation studies and as a platform to develop specific plug-ins, due to its modularity nature. Once the solution is evaluated, data processing and plotting of the results was done using MATLAB [9]. In case results were not satisfactory, based on the feedback, we retraced our steps, either fixing some aspects or bugs of the solution or even reconsidering the metrics of the problem. Finally, results were published and the whole processed was started again with a new innovation for a previous goal, or a new idea.

Figure 1.1: Methodology

1.4 Outline

This document is divided into nine chapters, including the current one, plus one appendix. Chapters are group into three differentiated parts, highlighting the main topics covered by this thesis. Furthermore, two additional parts present the introduction and closing discussions. Each chapter focuses on a specific subtopic.

Part I resents a general overview of this thesis (see Chapter 1), while the essential technical background and state of the art necessary to properly understand this document is covered in Chapter 2.

In Part II, we present contributions related to network planning. First, a multilayer planning algorithm to ensure not only 100% survivability of network but also guaranteeing a maximum end-to-end latency even in a failure scenario (see Chapter 3) is presented. Then, we analyze different schemes for elastic spectrum management on multihour filterless metro networks (see Chapter 4). Lastly, we devise a planning strategy to overcome the drawback of WaveBand-ROADMs in metro networks (see Chapter 5).

In part Part III, we focus on the dynamic operation of multilayer networks using SDN techniques. First, we present an experiment, using Net2Plan as an off-platform application for an SDN controller with the goal of orchestrating OpenFlow-based networks (see Chapter 6). Second, we develop an active-stateful multilayer path computation element to orchestrate multilayer service provisioning and restoration in IP/MPLS-over-WDM networks (see Chapter 7).

In part Part IV, we discuss the dynamic operation and VNF placement on 5G access networks. Particularly, Chapter 8 focuses on a joint orchestration for efficient service chain deployment in dynamic mobility scenarios.

Finally, in Part V, concluding remarks about this thesis are presented, as well as possible extensions and further work (see Chapter 9). After that, Appendix A closes this document, showing the various reference network topologies used in different works throughout this Ph.D. thesis.

Chapter 2

General Context

2.1 Technological Background	7
2.1.1 Multilayer Optical Networks	7
2.1.2 Software-Defined Networking	10
2.1.3 Network Function Virtualization	13
2.2 Evaluation Methodology	15
2.2.1 Network Planning	16
2.2.2 Network Simulation	16
2.2.3 Net2Plan	17

2.1 Technological Background

2.1.1 Multilayer Optical Networks

Due to the vast number of technologies and requirements involved in communication networks, classifying them can be difficult. From local area networks (LANs) to trans-oceanic wide area networks (WANs), they can be classified depending on the geographical extent. Nonetheless, regardless of the technology involved, cost-efficiency is a key issue common for all these networks, and this problem needs to be addressed from the early planning stages to the regular operation of the network. Therefore, the basic structure of the networks must be known beforehand. In this regard, depending on the perspective, all networks can be deconstructed on two ways: *partitioning* and *layering* [10].

Partitioning helps to better differentiate domain separation according to ownership and/or spatial coverage (e.g., autonomous systems in the context of Internet protocol (IP) addressing). Following this categorization, a geographical partitioning into access, metro, and core/backbone networks, or even a hybrid of them (e.g., metro-access or metro-core), is generally used, as depicted in Figure 2.1 (adapted from [11, Section 1.2]). These network

types differ not only in the technology involved but also in the services provided. Traditionally backbone optical network nodes are composed of devices called Reconfigurable Optical Add/Drop Multiplexers (ROADMs); these elements allow the establishment of optical circuits that span a path of multiple fibers and nodes. Each optical circuit, or lightpath, is perceived as an individual IP link (from origin to destination) in the upper layer.

Figure 2.1: Network hierarchy classification

Unfortunately, partitioning a network cannot be used to reveal the type of services and data that are transmitted, or the particulars about how the transmission is achieved. On the other hand, using a layering perspective, not only transmission equipment, but also encoding and protocols can be defined.

The basic multilayer model that has been considered throughout this Ph.D. thesis is presented in Figure 2.2. This model is comprised of two layers: the *optical layer* and the *client/aggregation layer*. Nowadays, the aggregation layer mainly comprises a stack of IP and Optical Transport Network (OTN); where IP provides finer granularity (per-packet level) and OTN provides coarse granularity and additional grooming and multiplexing capabilities, intending to increase the efficiency of optical resources. In this thesis, we will assume OTN is just an adaptation layer of negligible impact on network design and operation.

2.1.1.1 Continuity and Contiguity Constraints

In optical networks, lightpaths (optical circuits) are allocated into a contiguous set of subcarriers (or slices) that are continuously available on every link along the path. Unfortunately, the technological constraints imposed by the spectrum contiguity (all slices must be contiguous) and the spectrum continuity (occupied slices must be the same throughout the fiber path), entail a problem called spectrum fragmentation. During network operation, lightpaths connections of different spectral bandwidths are set up and torn down, leaving the spectrum divided into small-sized blocks of available slices (vertical fragmentation), which may also be misaligned among neighbor links (horizontal fragmentation). Spectrum fragmentation degrades performance, i.e., the links may have unused bandwidth but the lack of contiguity and/or continuity in the spectrum may lead to blocking

2.1. Technological Background

Figure 2.2: Example of a two-layered network

requests.

The fragmentation problem is illustrated in Figure 2.3 (adapted from [12]). A connection is to be established between nodes A and C , requiring four spectrum slices. The link $A \rightarrow B$ has four available slices, but they are not contiguous due to vertical fragmentation. Link $B \rightarrow C$ has four contiguous slices, however, they do not match with the four contiguous slots in link $A \rightarrow B$, which is known as horizontal fragmentation.

Figure 2.3: Spectrum fragmentation in optical networks

2.1.2 Software-Defined Networking

From a functional point of view, computer networks can be divided into three different planes: the data, control, and management planes (see Figure 2.4, adapted from [13]). The data plane refers to the networking devices, which are responsible for an efficient data switching and forwarding. The control plane represents the protocols used to populate the forwarding tables of the different data plane elements. The management plane comprises the software services and applications used to remotely monitor and configure the control functionality, e.g., Simple Network Management Protocol (SNMP)-based tools [14]. The management plane is responsible for defining network policies, where the control and forwarding planes are in charge of installing and enforcing them, respectively.

Traditionally, the control and data planes have been intrinsically coupled and embedded in the same networking devices (specially in IP networks), and as a result, the whole structure is highly decentralized. In the early days, this was considered an important part of the Internet design, and the best approach to guarantee network resilience, which is of the utmost importance. In fact, this approach has been quite effective in terms of network performance, with a rapid increase of line rate and port densities.

However, as a consequence, traditional networks are rigid and complex to orchestrate. These two characteristics are largely responsible for a vertically integrated industry where innovation is difficult. Today's networks are prone to errors, and misconfiguration is extremely common. For example, in 2005 more than a thousand configuration errors were reported in border gateway protocol (BGP) routers [15]. A single misconfigured device may result in undesired network behavior, including packet losses, forwarding loops, unintended paths, contract violations and so on.

To support network management, a small number of vendors offer proprietary solutions of specialized hardware, operating systems, and control programs (network applications). However, such solutions are commonly only compatible with the devices sold by the same vendors. If network operators want to achieve some sort of interoperable management, they have to acquire and maintain different management solutions and the corresponding specialized teams. The capital expenditure (CAPEX) and operational expenditure (OPEX) of building and maintaining a network infrastructure is significant and with long return on investment cycles, which prevent innovation and addition of new features and services (e.g., access control, load balancing, traffic engineering, and so on).

The birth of SDN is tied to OpenFlow and the seminal work done by Stanford University. The original definition refers to a network paradigm, where the forwarding state in the data plane is decoupled from the control plane, which remotely controls the former. Nonetheless, many times the term has shifted from the original definition by referring to anything that involves “software” in network management. A more strict definition based on four pillars was provided by [13]:

1. Decoupling of control and data planes. Control functionality is removed from network devices, those become mere forwarding elements.
2. Flow-based forwarding decisions, instead of destination-based. A flow can be defined by a set of field values acting as a match/filter criterion and a set of actions or instructions. This abstraction allows the behavior unification of different network

2.1. Technological Background

Figure 2.4: Planes of functionality in networking

devices, such as routers, switches, firewalls, and middleboxes [16].

3. External control logic. Since the control plane is decoupled from the devices, an SDN controller or Network Operating System (NOS) is an external entity that provides control logic. The NOS is a software platform that runs on a general-purpose server and provides the essential resources and abstractions to facilitate the programming of forwarding devices based on a logically centralized, abstract network view. Its purpose is therefore similar to that of a traditional operating system.
4. Off-platform programmability. The network is programmable through software applications running on top of the NOS that interacts with the underlying data plane devices. This is a fundamental characteristic of SDN, considered as its main value proposition.

It is worth noting that the centralization of the control logic offers several additional benefits. It becomes simpler and less prone to errors to modify network policies through high-level languages and software components, compared with low-level device-specific configurations. Also, a control program can automatically react to spurious changes of the network state and thus maintain the high-level policies intact. And lastly, a centralized controller with a global knowledge of the network state simplifies the development of more sophisticated networking functions, applications, and services.

2.1.2.1 Controller vs Network Operating System vs Orchestrator: What is the Difference?

Operating systems have traditionally provided high-level abstractions for accessing lower level devices and manage the concurrent access to the underlying resources (e.g., hard drive, volatile memory, CPU). These functionalities are key enablers for making easier the life of system and application developers as well as to contribute to the evolution of programming languages and applications. On the other hand, until recently, networks have been managed and configured using lower level, device-specific instructions sets

and closed proprietary NOSs (e.g., Cisco IOS, Juniper OS). Furthermore, the very idea of operating systems that abstract device-specific characteristics and provide common functionalities is still mostly absent in networks.

The emergence of SDN aims to facilitate network management and ease the burden of solving networking problems by taking advantage of the logically centralized control offered by a NOS. As with traditional operating systems, a crucial aspect of a NOS is to provide abstractions, essential services, and common application programming interfaces (APIs) to developers. Generic functionalities such as network state and topology information/discovery, device discovery, and dissemination of network configuration can be services provided by a NOS. Such systems can arguably create a new environment capable of fostering innovation at a faster pace by reducing the inherent complexity of creating new network protocols and network applications. With the help of NOSs to define network policies, developers no longer need to care about low-level details or device-specific configuration among routing elements. Such platforms should help to create new environments capable of fostering innovation at a faster pace by reducing the inherent complexity of creating new network protocols and network applications.

An SDN controller or NOS is a critical element in the SDN architecture (see Figure 2.5, adapted from: [17]) as well as a key component for the control logic (applications) to generate the network configuration based on the policies defined by the network operator.

Figure 2.5: Typical SDN architecture

Besides, in recent years the term *SDN orchestrator* has gained a lot of traction; the question then remains, what is the difference between a controller and an orchestrator? The abstraction of functional planes in Section 2.1.2 defines the forwarding plane as the one responsible for forwarding data, as dictated by the control plane. The SDN controller is the entity for deciding how and where to forward the traffic. Conversely, the (network) orchestrator is a management plane application in charge of the life-cycle management of the whole system, which includes not only the forwarding, but provisioning, monitoring and maintenance, usually of both network and services running on top. In fact, the orchestrator can be often considered as part of the network management system (NMS)/operational support system (OSS), and may support some fault, configuration, accounting, performance and security (FCAPS) capabilities.

2.1. Technological Background

2.1.2.2 OpenFlow

As aforementioned, the emergence of SDN made possible the appearance of APIs to simplify the implementation of network services. OpenFlow [18] is one of the early implementations of these open interfaces, developed to standardize information exchange between the control and data plane. In OpenFlow, the hardware architecture is as follows:

1. A flow table containing flow entries comprising match rules and actions to take on active flows.
2. A transport layer protocol that communicates securely with a controller/NOS about new entries that are not currently on the flow table.

Full OpenFlow switches possess no legacy features or on-board control, and completely rely on a controller/NOS for forwarding decisions. On the other hand “hybrid” switches support OpenFlow in addition to other traditional protocols. Nowadays, most commercial switches available are hybrid. An OpenFlow switch comprises one, or several, flow tables that perform packet lookup and forwarding. Each flow is in turn composed of these entities:

1. Header/Match fields. These contain information found in the packet header, ingress port, and also metadata used to match incoming packets.
2. Counters. Statistics such as received packets, number of bytes and duration of the flow are collected by counters.
3. Actions. After a match, instructions or actions dictate how to handle matching packets. E.g., forwarding the packet to a specific port or drop it.

Figure 2.6 (source: [19]) shows a typical SDN architecture composed of network applications, controllers, OpenFlow switches and an example of flow table. The specification describes two different management approaches regarding how to populate flow tables: proactive and reactive. Reactive mode occurs when incoming packets do not match any of the existing flow entries. These packets are sent to the controller, which takes a decision based on its programming and then installs rules back into the switch. Conversely, the controller may install flow entries proactively, according to user policies, before packets arrive. The reactive approach is a more powerful mode because the controller can implement any kind of logic that cannot be represented into flow tables. On the other hand, proactive mode yields better performance due to the fact it avoids the extra latency added by sending to the controller the first packet of the flow. In practice, applications may implement a combination of both modes, where part of the traffic is handled proactively and part is handled reactively. This mixed mode would provide no additional latency forwarding for particular flows while still preserving fine-grained traffic control for the rest of the traffic.

2.1.3 Network Function Virtualization

Traditionally, service provisioning in the telecommunications industry was based on network operators deploying proprietary and specific devices and equipment for each function [20]. As a consequence, this prevents an efficient operation, service addition, and network upgrades. To address this issue and reduce CAPEX and OPEX, virtualiza-

Figure 2.6: OpenFlow Flow table and typical SDN architecture

tion emerged as a solution, decoupling the networking software applications from their supported hardware, allowing network services to be implemented as software, and be deployed in generic purpose hardware [21]. Taking advantage of virtualization technologies, European Telecommunications Standards Institute (ETSI) proposed Network Function Virtualization (NFV) as a new paradigm, to virtualize network functions that previously were provided by proprietary and dedicated hardware [22]. The decoupling of network functions from the underlying hardware appliances, this new paradigm provides flexible provisioning of virtualized network functionalities on top of an optimal shared physical infrastructure, while addressing the problems of operational costs tied to the management and control of closed proprietary appliances.

2.1.3.1 NFV Architecture

Figure 2.7 (source: [23]) depicts the NFV architecture proposed by ETSI, comprising three elements: Network Function Virtualization Infrastructure (NFVI), virtual network functions (VNFs) and NFV Management and Orchestration (MANO) [24].

The NFVI refers to both hardware and software resources, in which VNFs are deployed. This includes commercial-of-the-shelf hardware, storage, and network nodes and links; these provide computing, storage and network connectivity to the VNFs respectively. The virtualization layer (using a hypervisor) provides an abstraction view of these resources, decoupling the “virtual resources” from the underlying physical resources. In data centers, computing and storage resources may be represented as one or more virtual machines (VMs) while virtual networks are made up of virtual links and nodes. Also, a virtual node is considered software that possesses hosting or routing capabilities.

A network function (NF) is a functional block that has well defined external interfaces and well-defined functional behavior [25]. Examples of NFs are firewalls, gateways, DHCP servers and so on. A VNF is a virtual implementation of a NF that is deployed over virtual resources and from the perspective of the user, they should be indistinguishable and have the same performance.

According to ETSI, the MANO framework provides the functionality required to pro-

Figure 2.7: Network function virtualization architecture

vision VNFs, their configuration and the management of the virtual infrastructure they run on. This includes managing the lifecycle of VNFs, and the databases that are used to store the information and data models that define properties of functions, services, and resources. Additionally, the framework defines interfaces that can be used for the communication between the different components of the NFV MANO, as well as coordinating traditional network management systems such as OSS and Business Support Systems (BSS) to allow the management of both VNFs and NFs running on legacy equipment. as to allow for management of both VNFs as well as functions running on legacy equipment.

2.1.3.2 Service Chaining

Service chaining is an important model for network service providers, in which NFV plays an important role [21]. A service function (SF) is a NF that is responsible for specific treatment of received packets, while a Service Function Chain (SFC) defines an ordered set of abstract service functions and ordering constraints that must be applied to packets and/or frames and/or flows selected as a result of classification [26].

2.2 Evaluation Methodology

Several options can be used to evaluate multilayer optical networks: software simulations, real-world/emulator-based testbeds, analytical models... Due to the inherent complexity of some network architectures, it may be difficult to develop analytical models via Mixed Integer Linear Programming (MILP), and since non-polynomial is required to solve this type of problem, the use of heuristics was widely used in this Ph.D. thesis.

Finding an optimal scenario, performance or deployment is fundamental in every topic addressed in this thesis either in network planning or orchestration. In this regard, when

the term “optimization” is used, two are the fundamental objectives: (i) to properly understand the behavior and dynamics of the different elements comprising a network, and (ii) to identify the key points where performance can be optimized respect to previous scenarios. Therefore, instead of validating our proposals by simply comparing them to MILP solutions for scaled-down problem instances, we benchmarked the performance of our algorithms to previous proposals, in order to verify whether we were able or not to outperform their results.

2.2.1 Network Planning

The purpose of network planning is to focus on the details pertaining the accommodation of traffic carried by the network. This includes the decision of how a demand should be routed and protected, and what wavelength of the spectrum is going to be allocated to carry it. Planning can be classified depending on the time-scale, for example: real-time, medium-term and long-term planning. Long-term planning involves designing or expanding the network to meet the requirements of a specific period of time, which may span from months to years. Since there is enough time between the planning process and the provisioning of services, additional capacity and equipment required can be deployed. This typically takes place before the network is properly deployed, and there is generally a large set of demands to be processed at once. Therefore, the main focus of this planning is finding the optimal strategy for accommodating this set of demands. On the other hand, the problem associated with this phase is the “uncapacitated design”, i.e, the fact that there is a lack of resource constraints. Once the network capacity has been installed and it is operational, medium-term planning refers to the accommodation of the incrementally added traffic. In this phase, the main problem is to optimally allocate demands taking into account the installed capacity constraints to maximize the network throughput; this is called “capacitated design”. Lastly, real-time planning means there is little time between planning and provisioning, and demands arrive randomly; i.e., the arrival and departure time, as well as the requested capacity are unknown beforehand. The objective of this phase is to carry each demand using whatever equipment is deployed and available in the network while minimizing the blocking probability. This thesis will deal mainly with long-term and real-time planning.

2.2.2 Network Simulation

The network simulations performed on this thesis are based on the concept of Discrete Event Simulation (DES) Each event represents an action (or a set of) that takes place in a certain time instant. Some of the expected actions could be a new lightpath request, a link failure, and so on.

A typical DES-based system comprises the following elements, as listed in [27]:

- *System state*. Set of variables and/or data structures representing the state of the network at a given time, this set is modified by the evens of the simulation.
- *Event list*. Represents the list of future events, in ascending order according to their occurrence time.

2.2. Evaluation Methodology

- *System clock.* A variable that tracks the time in the simulated scenario, usually differing from the real-world time.
- *Statistics collector.* Retrieves different metrics describing the state and performance of the network, usually stored in files for further processing.
- *Event procedure.* Set of actions that should be executed upon the arrival of a particular event.
- *Initialization.* The procedure used at the beginning of the simulation.
- *Report generator.* It allows the generation of reports based on the output of the Statistics collector.
- *Main program.* It is responsible for the management of all previously described elements.

2.2.3 Net2Plan

Net2Plan is an open-source network planning tool¹ developed in Java and publicly and freely available to download from its website [28].

Net2Plan [8] evolved from the MatPlanWDM [29, 30]. This software was intended for designing and planning multilayer wavelength division multiplexing (WDM) networks and was implemented as a MATLAB toolbox, including a graphical user interface (GUI). Eventually, Net2Plan was created as part of the Ph.D. thesis of Dr. José Luis Izquierdo Zaragoza [12] in collaboration with his advisor Prof. Pablo Pavón Mariño; it is an open-source technology-agnostic multilayer network planning and in-operation simulation tool.

Nowadays, Net2Plan is used not only by industry and academia but also for teaching courses at university. All of this in addition to the growing repository of network planning resources [28, 31] has made Net2Plan into a useful and powerful tool. This software was fundamental in the development of this thesis, used in Part II to evaluate network planning algorithms. In Part III we prepared several extensions to integrate Net2Plan into SDN deployments. And lastly, in Part IV, the simulation tool was widely used.

2.2.3.1 Features

In Net2Plan networks are represented as a set of nodes, links, routes, traffic demands and network layers; all represented as abstract elements without relying on any particular technology. Also, it is specifically designed to allow users to integrate their own developed algorithms as Java classes implementing particular and public interfaces. Once an algorithm is developed within the Net2Plan framework, it can be applied to any network design. A large set of built-in examples are included (together with the source code), many of them thoroughly described in the examples section of the website [28]. The Net2Plan philosophy enforces software reutilization so that algorithms can be modified or reused as parts of others. In addition, it is intended to become a repository for network planning algorithms in any technology. As a teaching resource, students can use Net2Plan to examine the planning of particular technologies. Net2Plan is also a valuable resource for network planning courses that concentrate on network optimization concepts

¹Net2Plan is licensed under the BSD-2 License

and transversal network planning skills, avoiding an in-depth study of technologies that appear different as proposed by other authors [32].

Net2Plan provides both a GUI and a CLI. In either mode, Net2Plan currently provides three different tools²:

- Offline network design & Online Network Simulation.

The offline network design aims to evaluate the network designs generated by built-in or user-defined offline network design algorithms, deciding on aspects such as the network topology, the traffic routing, link capacities, protection routes and so on. If needed, those algorithms based on constrained optimization formulations (e.g. ILPs) can be fast-prototyped using the open-source Java Optimization Modeler library (JOM), to interface to several external solvers such as GLPK, CPLEX, XPRESS or IPOPT.

The online simulation tool allows the evaluation of network recovery schemes, connection-admission-control (CAC) systems, or dynamic provisioning algorithms for time-varying traffic. At the end of the simulation, some metrics (including either predefined metrics or custom ones) are presented.

- Traffic matrix generation.

Assists users in the process of generating and normalizing traffic matrices i.e. following random models found in the literature.

- Reporting.

Net2Plan allows the generation of built-in or user-defined reports, from any network design. The report tool is integrated within all the previous functionalities so that it is possible to create reports collecting performance measures in any of these aspects.

We recall that every algorithm, report, event generator, and event processor in Net2Plan can be either built-in (from the repository [28]) or user-made by implementing open interfaces in Java. We refer the reader to [28] for fully-detailed information regarding Net2Plan functionalities.

Next, some information about tools for offline and online studies are briefly presented.

2.2.3.2 Offline Network Design

This tool assists users in the process of offline network design and planning. In practical network design, different variables can be involved: network nodes, network links, link capacities, traffic routing, traffic demands, and so on. Generally, offline network design problems receive some of this information as input parameters (e.g. traffic demand, and network topology) and try to optimize others (e.g. capacities in the links and traffic routing), according to a performance merit of interest. The word “offline” here means that all the variables in the network plan are supposed to be deterministic values, which also do not change along time. For example, offered traffics are assumed to be constants representing the average traffic volumes, although, in reality, the traffic can fluctuate around this average according to statistical patterns.

The workspace of the offline network & simulation tool design window is divided into

²This information is referred to version 0.6.3, which was the latest one used by the candidate

2.2. Evaluation Methodology

two areas: input data, algorithm/simulation execution and reporting (right area) and plot area (left area). In Net2Plan, network designs can be completed incrementally, modifying the current network design, which the user can inspect visually in the “Network” panel and the “View/Edit network state” tab. When the tool starts, the current network design is empty. Then, the user can modify it (i) using manual editions in the “Network” panel or the “View/Edit network state” tab, (ii) loading existing designs and/or traffics demands, (iii) applying offline design algorithms (that take the current network design, and return a modified design becoming the current design, see Figure 2.8a).

2.2.3.3 Online Simulator

In a real-world environment, network conditions vary during its operation, according to different phenomena. Failures in nodes and links, the establishment of new virtual circuits, or variation in traffic volumes are some examples. In this case, users could be interested in analyzing, using the event-driven simulation, how their networks react to those changes and how their designs are consequently adapted for it.

The interface is similar to the offline network design tool, with three basic differences: (i) users cannot modify manually the design, (ii) users should incorporate an event generator and an event processor instead of a design algorithm in the “Online algorithm” tab (see Figure 2.8b), and (iii) after finishing the simulation a report with some metrics (including either predefined or own-developed metrics) is shown in “Simulation report” subtab).

(a) Offline algorithms

(b) Online simulation

Figure 2.8: Offline network design & online simulation tool

Part II

Network Planning

Traffic Survivability and Latency Awareness in Multilayer Network Design

3.1	Network Planning: When Survivability is not Enough	23
3.2	Case Study	25
3.2.1	Algorithm Design	25
3.2.2	IP-over-WDM Module and Recovery Schemes	28
3.2.3	Latency Computation	29
3.2.4	Testing Scenario	29
3.3	Conclusions and Further Work	33

As the first contribution under Part II, this chapter presents some foundations of network optimization in the context of multilayer optical networks. The underlying network architecture used here is the plain vanilla meshed IP-over-WDM solution deployed for the last two decades, where IP routers are interconnected via optical circuits switched by ROADMs without Optical-Electrical-Optical conversion. Without loss of generality, next chapters will address different variants on the optical layer, being the IP layer deliberately excluded, when linear topologies (either ring or horseshoe) are in place.

3.1 Network Planning: When Survivability is not Enough

There is no doubt that nowadays the telecommunication sector is undergoing a major transformation. The emergence of over-the-top (OTT) companies and services like Netflix or Google Stadia are challenging traditional operators in many different ways but, most importantly, they are forcing them to evolve their current rigid legacy infrastructures towards more simplified, automated, dynamic and agile architectures, getting rid of static configuration processes [33] and peak-based capacity dimensioning [34]. As a matter of fact, adding more capacity to the network is not enough, and they should guarantee a

good quality of service (QoS)/quality of experience (QoE) for services not under their full control.

The main QoS/QoE metrics for (real-time) multimedia services, like those provided by OTTs, are latency and jitter. The former is the time the information needs to reach end-users from the content generator. The latter is the variance of latency. In this new market scenario, these metrics must coexist with other typical figures in service-level agreements (SLAs), like availability levels [35] (e.g., five-nines). Unfortunately, this topic has received limited attention from the community [36].

In IP-over-WDM networks, the end-to-end latency of an IP flow depends not only on the sequence of IP links (lightpaths) traversed but also on the length of each one of these lightpaths through their physical path. However, IP and optical layers are operated by different personnel and departments, and IP-optical mapping is not a trivial task. Traffic survivability is currently assured through a dual-plane approach, in which two network copies mutually protect each other (i.e., 1+1 protection), and the IP layer is over-provisioned [2]. On the other hand, multilayer restoration has been demonstrated to emulate survivability capabilities at a reduced cost [34]. Anyway, most of the existing approaches in the literature do not take control of the latency under failure states.

Several studies have demonstrated the benefits of a joint fault-tolerant design of both optical and IP in multilayer networks [34] in terms of costs; the main contribution of this work consists of improving the design phase to include both fault tolerance and latency-awareness, as an improvement and extension of a previous conference paper [37], with negligible effect in costs.

Although the use of Multi-Protocol Label Switching with Traffic Engineering extensions (MPLS-TE) is a rising trend, many networks still use Open Shortest Path First (OSPF)-Equal-Cost Multi-Path (ECMP) in the IP layer [38]. This type of routing, while not possessing traffic engineering or individual path control, requires a meticulous and careful design due to weight link tuning [39], and drastic variations in flow routing because of topology changes, i.e. up-to-down/down-to-up single link events.

In this chapter we present a planning algorithm for multilayer IP-over-WDM, guaranteeing not only survivability but end-to-end latency awareness even in case of failure. Also, a case study will focus on analyzing the network cost and throughput for different strategies, and the impact that imposing latency constraints has on those metrics. The three recovery methods considered in the case study are:

1. *1+1 optical protection* followed by IP restoration, where each IP link is realized by two shared-risk group (SRG)-disjoint lightpaths.
2. *IP-only restoration*, where failed lightpaths are not recovered by the optical layer, and just OSPF reroutes the traffic over the surviving IP layer.
3. *Optical-followed-by-IP restoration* (or multilayer restoration), which involves cooperation between the two layers. Here, the optical layer attempts first to restore failing lightpaths. After that, the IP/OSPF layer reroutes the IP traffic in the surviving lightpaths.

3.2 Case Study

3.2.1 Algorithm Design

The aim of the proposed multilayer algorithm is to create minimum cost IP-over-WDM designs where the 100% of the IP traffic survives to a set of predefined failure states, whereas guaranteeing the end-to-end latency at the IP level will not exceed a given threshold in any possible network state, either faulty or failure-free.

The input data of the algorithm are:

- IP-over-WDM topology ($G(N, E)$), comprising a set of multilayer-capable nodes ($n \in N$) interconnected via unidirectional fibers ($e \in E$)
- A set of arbitrary-defined SRGs ($r \in R$), representing the set of possible failures. Each SRG represents a reasonable risk we want to be tolerant to, and comprises the set of links and/or nodes that simultaneously fail when such risks happen (e.g., natural disaster). Without loss of generality, we assume the network will remain connected after any single SRG failure (i.e., a physical path can be found between each node pair), so that guaranteeing 100% survivability is always possible. The set of network states ($s \in S$) is composed of the failure-free state and all single SRG possible failures.
- A set of end-to-end IP demands ($d \in D$), representing the end-to-end traffic demands between node pairs.
- Maximum end-to-end latency value (L_{max}) to be met by any IP demand ($d \in D$) under any network state ($s \in S$). Only propagation delays at the traversed lightpaths are considered for computing the end-to-end latencies.

In order to analyze traffic survivability and latency constraints, the algorithm considers three different recovery mechanisms in case of fiber failure: 1+1 optical protection followed by IP restoration, IP-only restoration, and optical-followed-by-IP restoration (also known as multilayer restoration [40]). Hence, we are able to dimension the IP layer according to the expected behavior of the network.

As aforementioned, the routing in the IP layer is governed by OSPF-ECMP according to the current standards, having all links a weight equal to one. At the optical layer, it is assumed WDM technology, with no wavelength conversion and no need for intermediate regeneration.

Multilayer network design variants are $\mathcal{NP}$ -complete problems, and we resort to a heuristic algorithm in our algorithm design, based on the Greedy Randomized Adaptive Search Procedure (GRASP) [41] scheme. GRASP is a well-known iterative meta-heuristic, composed of two differentiated phases per iteration: (i) construction and (ii) local search. In the construction phase, a feasible solution is constructed using a greedy algorithm. In this case, we consider as a feasible solution a network design that carries all the offered traffic, ensuring 100% survivability in case of single SRG failure, and meeting a maximum end-to-end latency in all states. In case a solution cannot be found, i.e., no more lightpaths cannot be allocated and therefore traffic is blocked, the algorithm starts a new iteration. Once a solution is found, the second phase tries to improve the objective function using a local search algorithm, trying to reduce the cost by minimizing the total number of

transponders. After finishing all iterations, the best solution found so far is returned.

The algorithm is composed of several hierarchical modules (see Figure 3.1). Each module is tasked to take pre-defined actions and delegating into other modules those events outside its scope. The main module contains the aforementioned GRASP scheme. The IP-over-WDM module is used to apply routing and wavelength assignment (RWA) and recovery mechanisms as it would be in a real-world scenario. Addition or removing of lightpaths is done by the WDM module, and the IP module is tasked with recalculating OSPF-ECMP rules at the IP layer each time a new link is added or removed.

For example, the main module sends an event to the IP-over-WDM module (e.g., “Add-lightpath” or “SRG-failed”), the latter reacts by taking some actions or sending new events to the WDM and/or IP module depending on the type of event received. This procedure allows us to generate offline planning algorithms by reusing code from existing online algorithms, for the purpose of dynamic lightpath allocation, setting up recovery schemes, and so on.

Figure 3.1: Algorithm modules hierarchy

The construction phase starts with the input data described above, and its pseudo-code is presented in Algorithm 3.1. Initially, the network is empty, and therefore all traffic is blocked. In each iteration of the main loop, one lightpath is added, and the traffic is routed by the IP layer. The `evalConstraints()` method, receives a design, and evaluates whether the fault-tolerance and end-to-end latency constraints are met. Then, it returns the set of IP demands with blocked traffic (BT_D), for which no IP path exists, the set of IP links with over-subscribed traffic (OT_{EIP}), and the set of IP demands which violate the maximum latency constraint ($maxL_D$), storing the worst-case among all states for each of these variables. The construction phase ends by leaving the main loop, when (i) no constraint is violated, that is, when BT_D , OT_{EIP} and $maxL_D$ become empty, or (ii) no additional lightpath can be setup (unfeasible solution) leading to a new iteration of the main algorithm.

Delving into the `evalConstraints()` method, we iterate over all possible network states ($s \in S$) to check the fault tolerance and latency constraints in each one of them. To do so, since every failure state is modeled as an SRG, an “SRG-failed” event is sent to the IP-over-WDM module, which reproduces the reaction of the IP-over-WDM network when transiting from a non-failure state to the failure state s . The reaction of each

3.2. Case Study

```

Input:  $G(V, E)$ ,  $D$ ,  $S$ ,  $L_{max}$ 
 $[BT_D, OT_{E_{IP}}, maxL_D] = \text{evalConstraints}()$ 
while  $BT_D \neq \emptyset$  or  $OT_{E_{IP}} \neq \emptyset$  or  $maxL_D \neq \emptyset$  do
    if  $BT_D \neq \emptyset$  then
        select randomly weighted an ingress-egress pair of nodes with blocked
        traffic
    else if ( $OT_{E_{IP}} \neq \emptyset$  and  $random\_number < \alpha$ ) then
        select randomly weighted an ingress-egress pair of nodes with
        over-subscribed IP links between them
    else
        select randomly weighted an ingress-egress pair of nodes with non-allowed
        latency between them
    end
    establish a new IP link (lightpath) between the selected node pair
     $[BT_D, OT_{E_{IP}}, maxL_D] = \text{evalConstraints}()$ 
end

```

Algorithm 3.1: Pseudo-code for the algorithm construction phase

module depends on the recovery mechanism considered (see Section 3.2.2). Then, the resulting state is checked, some internal metrics (those defined in the previous paragraph) are computed and the network is reverted to normal operation, i.e., no failure.

After the execution of the `evalConstraints()` function, if at least one constraint is violated (and thus the design is still invalid), we proceed as follows. First, we check the presence of pair of nodes with blocked traffic (BT_D), selecting among those a pair of ingress-egress nodes using a weighted random method, where each weight value is the amount of blocked traffic. In case there is no blocked traffic, if some link is over-subscribed we pick a random number and compare it to an input parameter α to determine if a new lightpath must be setup between oversubscribed links. The rationale behind this decision is to moderate the probability of adding too many lightpaths. If a new lightpath is to be added, a node pair is weighted randomly selected from those in $OT_{E_{IP}}$, using over-subscribed traffic as weight. Finally, if none of the previous options were selected, we check whether it exists any IP demand ($maxL_D$) exceeding the maximum end-to-end latency value. Again, a weighted random selection is performed using the excess latency (in milliseconds) as weight.

Once a node pair has been selected (through any of the conditions mentioned above), a new lightpath (IP link) is established between the ingress and the egress node. To allocate this new lightpath, we follow a fixed-alternate routing strategy along with a first-fit wavelength allocation. In other words, we have a pre-computed set of candidate paths between each node pair and try to find the shortest path, selecting the common free wavelength with the lower index. Due to the RWA constraints, this part of the algorithm might result in the inability to allocate a lightpath and therefore a feasible solution cannot be provided. This stops the construction phase.

Finally, the constraints are computed again, and the whole process is repeated until

a feasible solution is found (with no constraint violation) or no more lightpaths can be established (invalid solution). If a feasible solution can be found, then the local search stage is started.

The purpose of the local search phase is to improve the cost of a feasible solution coming from the first phase, removing as many IP links as possible while still satisfying the survivability and latency requirements. Algorithm 3.2 depicts the corresponding pseudo-code. First, all IP links are ordered in terms of their spare capacity in descending order (E_{IP}). Then, we iterate over them, assuming in each iteration the selected IP link is removed, and evaluating the consequences of such removal with `evalConstraints()`. In case no constraint is violated in any of the failure states, the IP link is effectively removed. This process is repeated until no more IP links can be eliminated.

Input: A feasible solution from Construction phase

```

repeat
  order all IP links by decreasing order of spare capacity (denoted as  $E_{IP}$ )
  forall IP link  $e_{IP} \in E_{IP}$  do
    mark  $e_{IP}$  as removed
     $[BT_D, OT_{E_{IP}}, maxL_D] = \text{evalConstraints}()$ 
    if  $BT_D = \emptyset$  and  $OT_{E_{IP}} = \emptyset$  and  $maxL_D = \emptyset$  then
      remove IP link  $e_{IP}$ 
      break
    else
      unmark  $e_{IP}$ 
    end
  end
until no IP link can be removed
  
```

Algorithm 3.2: Pseudo-Code for the Local Search Phase

3.2.2 IP-over-WDM Module and Recovery Schemes

We developed the IP-over-WDM module to support the three recovery schemes considered, using a bottom-up multilayer approach [42] (the same applied in [43]); that is, when an event is received, the main module first invokes actions from the WDM module to compute the reactions to the failure of the optical layer, and then calls the IP module to obtain the reaction of the IP layer to the surviving topology. The optical module actions depend on the selected recovery scheme, while the scheme computing the OSPF-ECMP reactions at the IP module is common for all recovery mechanisms (i.e., re-routing according to the surviving IP topology).

As for the WDM module, when using 1+1 optical protection followed by IP restoration, each new “Add-lightpath” event is treated realizing each IP link with two SRG-disjoint lightpaths. Therefore, in case of an “SRG-failed” event, no actions must be taken and traffic survivability is guaranteed as lightpaths were created as SRG-disjoint.

3.2. Case Study

For multilayer restoration, the WDM module is in charge of applying RWA strategies to allocate new lightpaths, and in case of failure event, to reroute as many failing lightpaths as possible over the surviving topology before the IP module recomputes the OSPF-ECMP rules.

Finally, for IP-only restoration, the WDM module takes no action at the optical layer when a failure occurs, since failing lightpaths are not restored. OSPF-ECMP rules are recomputed by the IP module.

3.2.3 Latency Computation

A thorough computation of end-to-end latency metric would involve taking into account the propagation delay, queuing delay, transmission delay, and packet processing times. Assuming the network is properly dimensioned, congestion episodes will be minimal and queuing delay will be minimized. Moreover, supra-gigabit/second line rates and packet processing at wire-speed leave the propagation delay as the main contribution to the latency in backbone networks [44]. Therefore, for the sake of simplicity, we will only consider propagation delay as the sole source of latency.

For this study, since IP/OSPF-ECMP is based on hop-by-hop routing, there is a need to obtain explicit end-to-end paths in order to compute per-flow propagation delays. Given an IP network, a set of IP demands, and IP link weights, we apply a path reconstruction method inspired by *Edmonds-Karp* algorithm [45] to compute end-to-end paths from ECMP forwarding rules. A more detailed explanation can be found in [31]. This computation is extensively used in `evalConstraints()` to retrieve the worst-case end-to-end delay among all paths carrying traffic for each IP demand and each failure scenario.

3.2.4 Testing Scenario

To test its performance, the algorithm was coded in Java and we devised a test scenario using the “Offline network design” tool of Net2Plan. We used three different well-known IP-over-WDM networks topologies composed of bidirectional fiber links and network sites (including multilayer equipment): NSFNet (see Appendix A.3), Internet2 (Appendix A.2) and Atlanta (Appendix A.4). The IP traffic for each network was generated using the population-distance model described in [46]. We assume each node is equipped with IP routers and ROADMs with 100 Gbps transponders, and 40 WDM channels are available per fiber. Lastly, we consider the propagation speed of light in the fiber to be 200,000 km/s and no need for signal regeneration between node pairs.

To standardize all tests, fiber lengths in all networks have been appropriately scaled to normalize the network diameter to 5000 km in all cases, which represents an end-to-end delay of 25 ms. We note that the network diameter is the longest (in this case, measured in kilometers) among all-pairs shortest paths.

We run the algorithm for each topology, (i) for different scaled versions of its seminal IP traffic matrix, with total offered traffic ranging from 500 Gbps to 16 Tbps in steps

of 500 Gbps, and (ii) different end-to-end latency limits: 50 ms, 62.5 ms, 75 ms, and unbounded (no limit).

One SRG is defined for each bidirectional fiber pair between each node pair. In this form, designs should be tolerant to single duct cuts, where the cut breaks simultaneously the two fibers in opposite directions in the duct [47]. This kind of failure may be caused by natural disasters, human error (civil engineering activities), or even represent the case of programmed maintenance (e.g., upgrade of optical line amplifiers). Note that the algorithm can easily accommodate multiple duct cuts, or other particular failure situations, by defining SRGs accordingly.

Once all the executions have finished we collect the total necessary number of transponders to establish the planned design for each of the recovery mechanisms. As occurs in practical deployments [11], transponders are supposed to be bidirectional, and transmission and reception lightpaths can be tuned at different wavelengths and follow different routes from the same end nodes. According to this, the number of transponders used in a node is the maximum between the number of all incoming and outgoing lightpaths to the node. As is customary in the literature, the number of transponders, which is tightly related to the number of necessary IP ports, is considered a good approximation of the overall network cost [48]. Finally, we collect the worst case of end-to-end latency for each IP traffic demand under any single SRG failure and the total throughput achieved with each recovery scheme and latency limitation.

Tables 3.1 to 3.3 show the overall throughput achieved for each of the networks under the three recovery schemes: 1+1 optical protection (1+1), IP-only restoration (IP-R) and optical-followed-by-IP restoration (Op-IP-R); and the different imposed latency limitations. Network throughput is computed as the sum of the traffic of the maximum IP traffic matrix for which a feasible survivable and latency aware design was found.

It is interesting to remark that a maximum allowed end-to-end latency of three times the normalized network diameter (75 ms) has no effect on the throughput respect to no limitation, no matter the selected recovery method. Even a limit of 62.5 ms (two times and a half the network diameter) of end-to-end latency has no noticeable difference, except on the case of 1+1 optical protection. The most consistent throughput is obtained using IP-only restoration, this may be due to the fact that in case of failure, no new lightpaths have to be allocated, removing the possibility of not having a sequence of links that meets the latency criteria.

Table 3.1: Overall throughput on NSFNet (in Tbps)

Latency Limit	1+1	IP-R	Op-IP-R
No limit	9.5	11.5	16.0
75.0 ms	9.0	11.5	16.0
62.5 ms	6.0	11.5	15.5
50.0 ms	N/A	7.0	6.0

Regarding the cost in terms of the necessary number of transponders, Figure 3.2 shows a comparison for each network between a design with no latency limit and a maximum limit of 62.5 ms for each of the three recovery schemes. Each value represents the total cost for a valid design which ensures 100% survivability.

3.2. Case Study

Table 3.2: Overall throughput on Internet2 (in Tbps)

Latency Limit	1+1	IP-R	Op-IP-R
No limit	8.0	7.0	10.0
75.0 ms	8.0	7.0	10.0
62.5 ms	8.0	7.0	10.0
50.0 ms	8.0	7.0	9.5

Table 3.3: Overall throughput on Atlanta (in Tbps)

Latency Limit	1+1	IP-R	Op-IP-R
No limit	5.5	8.5	9.5
75.0 ms	5.5	8.5	9.5
62.5 ms	3.0	8.5	9.5
50.0 ms	N/A	8.5	6.0

Comparing the three recovery schemes we can draw several interesting conclusions. First, there is a consistent ranking from highest to lowest cost for any given offered traffic in terms of transponder requirements: 1+1 optical protection, followed by IP-only restoration and optical-followed-by-IP restoration. This behavior is expected since 1+1 optical protection realizes each IP link as two different SRG-disjoint lightpaths, thus requiring two pairs of transponders. On the other hand, using IP-only restoration entails that additional lightpaths must be allocated before network operation to ensure no traffic is disrupted in case of failure, causing a network over-dimensioning. Finally, optical-followed-by-IP restoration is able to reroute failing lightpaths over the surviving physical topology and therefore some failures are unnoticed to the IP layer, being the trade-off a moderate over-dimensioning.

Second, from the point of view of a fixed number of transponders, the most efficient recovery scheme is optical-followed-by-IP restoration which achieves the maximum throughput compared to the two other alternatives.

Comparing for each recovery scheme (i) the lack of latency constraining to (ii) an imposed limit equivalent to 2.5 times the propagation delay of the network diameter, we can extract some interesting findings. First of all, the most observable difference is found in 1+1 optical protection. Using this kind of recovery while ensuring a maximum end-to-end latency at the IP layer leads to a notable increase in the number of transponders, as well as a much reduced total throughput (being the exception the Internet2 network). As for optical-followed-by-IP restoration and IP-only restoration, there is almost no increase in the number of transponders or decrease in the overall throughput (although in some particular cases it may seem a lower cost when imposing a latency limit, this may be caused to the random nature of the algorithm). It is safe to say that guaranteeing a maximum end-to-end latency on IP traffic without a perceptible increase in cost is always desirable.

Figure 3.3 provides a more detailed view of end-to-end latency behaviors in the network. It illustrates the histograms for worst-case (including failure states) latency distribution for each network for a given offered traffic of 3 Tbps. Each figure contains two

Figure 3.2: Total number of transponders for each topology (no latency limit versus a maximum limit of 62.5 ms)

sets of histograms, the upper one represents the worst-case end-to-end latency among all failure states for each recovery scheme, assuming no imposed limit to latency. The lower set contains the same information but limiting the maximum end-to-end latency to 62.5 ms. A vertical dashed lined highlights the 62.5 ms limit. We can easily observe that in the latency-aware scenario, the algorithm succeeds in making every flow meet the end-to-end latency constraint. In the case of unrestricted latency design, although a good proportion of IP flows meets the constraint criteria, many others exceed the maximum given latency, even by a large extent. This suggests that not considering latency limitations in the network design can easily produce large maximum latency violations.

Tables 3.4 to 3.6 show more detailed information about the proportion of IP demands and offered traffic exceeding maximum latency in case of an unrestricted design. We observe that without considering latency-aware limitations, a significant proportion of IP traffic will suffer from high latency, either during normal operation or in case of fail-

3.3. Conclusions and Further Work

Figure 3.3: Worst-case end-to-end latency distribution for each topology (no latency limit versus a maximum limit of 62.5 ms).

ure state. For instance, this impacts intensely on the optical-followed by IP restoration case. This scheme has been proven as the best solution both from lower cost and overall throughput perspective. However, its design should be especially careful, since not considering latency limitations may result in an unacceptable proportion of traffic exceeding the imposed maximum latency, as seen on NSFNet and Internet2 networks. Again, this phenomenon does not occur when applying latency-aware design thus supporting and validating the correct behavior of our proposed joint algorithm.

3.3 Conclusions and Further Work

In this chapter, we showed results that validate optical-followed-by-IP restoration (or multilayer restoration) as the winning recovery scheme both in terms of cost and overall throughput. These results for the joint fault-tolerant and latency-aware scenario addressed in this paper, are consistent with other previous results in less restricted scenarios. Still,

Table 3.4: Proportion of IP demands and offered traffic violating latency constraints on NSFNet (latency unaware)

Recovery Scheme	% IP Demands	% O.T	O.T (in Gbps)
1+1	19.23	19.46	584
IP-R	7.14	6.1	183
Op-IP-R	23.62	23.2	696

Table 3.5: Proportion of IP demands and offered traffic violating latency constraints on Internet2 (latency unaware)

Recovery Scheme	% IP Demands	% O.T	O.T (in Gbps)
1+1	4.16	6.13	184
IP-R	0	0	0
Op-IP-R	13.88	13.33	400

Table 3.6: Proportion of IP demands and offered traffic violating latency constraints on Atlanta (latency unaware)

Recovery Scheme	% IP Demands	% O.T	O.T (in Gbps)
1+1	18.75	9.5	285
IP-R	4.76	1.96	59
Op-IP-R	14.7	5.22	158

1+1-protected and unprotected lightpaths are still dominant in many telco networks, given the difficulty to perform and agile reconfiguration at the optical layer, required by lightpath restoration. In addition, even when possible, lightpath restoration times can be slow (e.g. tens of seconds to minutes) in current optical infrastructures, due to the need for channel equalization processes [34]. Moreover, OSPF re-convergence is in the order of dozens of seconds [49], so recovery time in case of optical-followed-by-IP restoration is at least as bad as IP-only restoration. On the other hand, 1+1 optical protection realizes each IP link as two disjoint lightpaths, so neither setup time nor re-convergence time is needed in case of single failure, being this method the fastest of the three. A good compromise between cost/throughput and recovery time would be the application of IP Fast ReRoute (IPFRR) techniques, which guarantee that OSPF reactions for rerouting IP traffic occur at a sub-second time. Therefore, further work includes the application of the aforementioned techniques to maximize IPFRR coverage [50] while maintaining end-to-end latency constraints for IP traffic.

Furthermore, it is of practical importance investigating algorithms that impose the latency restrictions selectively so some IP demands have different limits than others, prioritizing that way certain flows in terms of latency. In this scenario, the bottom-up restoration may not work as expected, and top-down approaches may become worth to analyze, provided that multilayer negotiation and coordination are not considered in the short-term. In fact, authors in [36] propose the application of IP-only restoration for sub-second recovery of high-priority services, followed by a combination of IP-optical mechanisms for the rest of the traffic.

Chapter

4

Spectrum Management in Filterless Metro Networks

4.1	Filterless Optical Networks	35
4.1.1	Transponder Technologies	37
4.1.2	Traffic Dynamicity	38
4.2	Case Study	39
4.2.1	Algorithm Implementation	40
4.2.2	Spectrum Allocation Approaches	41
4.2.3	Testing Scenario	42
4.3	Conclusions and Further Work	44

In the previous chapter, we discussed about meshed topologies, which are typical of transport/backbone networks (see Section 2.1.1). However, the adoption of ROADMs, as replacement of legacy optical ring technologies, is unaffordable for massive deployment in metropolitan areas. To overcome this issue, filterless technologies have gained momentum in the last years as a trade-off solution between cost efficiency and supported throughput. Filterless optical network are comprised of passive devices (i.e., splitters/couplers) instead of active optical switching. In this chapter, we present a scheme for planning filterless optical networks in a multi-hour realistic scenario.

4.1 Filterless Optical Networks

Filterless Optical Networks (FONs) have been introduced to target a significant cost reduction compared to existing optical networks. Optical nodes based on ROADMs and wavelength selective switches (WSS) (example depicted in Figure 4.1) provide agility in optical transport at the cost of incurring in additional CAPEX, which does not always compensate the benefits of agility. Leveraging coherent transponders (TXPs) and passive splitters/couplers, full-filterless nodes with the absence of active optical equipment (using

a *broadcast-and-select* architecture, see Figure 4.2) are deemed to be a cost-effective and reliable alternative to active optical nodes [51]. In FONs several optical trees are built over the same physical topology. In each of these trees, optical connections are based on *light-trees* and each optical signal reach all nodes. Thanks to coherent technology, each signal can be properly selected at the receiver (RX) of the corresponding destination node, following a *drop-and-waste* strategy [52].

Figure 4.1: 2-degree ROADMs with a broadcast-and-select architecture

Figure 4.2: Full-filterless 2-degree node architecture

Despite of cost savings and simple control and maintenance operations, FONs present several drawbacks: (i) there is a significant waste of spectrum resources since each signal occupies the entire associated fiber tree; (ii) simple topologies such as horseshoes or trees with no physical loops must be used to avoid power re-circulation and “lasing” effects of amplified spontaneous emission (ASE) noise; and (iii) careful control of the overall power entering each transponder is required, as all light channels enter each of them, provided that power levels must be above sensitivity.

As of today, filterless solutions have not been considered for large scale deployments [53]. The main motivation is that coherent technologies have been mainly adopted in

4.1. Filterless Optical Networks

backbone networks, where large ROADM-based mesh topologies are employed and the aforementioned filterless drawbacks have practically prevented the deployment of such drop and waste (DW) solutions.

However, the traffic growth occurring in metro networks is driving the replacement of traditional direct-detection 10 Gbps cards with more advanced solutions at 100 Gbps. In this context, where network operators usually adopt simple topologies (i.e., horse-shoes, rings and so on) over relatively short distances (up to 150 km) such transmission constraints become less critical, and coherent transponders over filterless transport may represent a suitable and cost-effective option. Nonetheless, the potential application of these solutions in the metro opens the way to additional and yet undiscussed design and technological aspects.

In this chapter, we explore filterless solutions in the context of planning metropolitan area networks, specifically addressing the following two aspects: (i) the impact of traffic dynamicity, which is significantly more intense than the one experienced in backbone networks, in the design of spectrum allocation (SA) solutions for filterless networks; and (ii) the availability of either a single or two tunable lasers within the coherent transponder, which may limit flexibility on bidirectionality.

4.1.1 Transponder Technologies

Coherent transponders are built in hardware as transceivers that handle within the same card both directions, i.e., transmitter (TX) and RX. According to the adopted technology and equipped hardware capabilities, transmission parameters may or not be configured in the same way both directions. For example, a connection between nodes A and B may or not be capable of supporting different central frequencies (CFs) from A to B (i.e., TX at A) and from B to A (i.e., RX at A). Such capability depends on the availability of either one or two tunable lasers and related electronic circuits within the card. If just one tunable laser is present, both TX and RX have to be operated over the same CF, i.e., the TX laser is also used as local oscillator at the RX side. Instead, if two tunable lasers are available in the same card, TX and RX can be operated over independent frequencies.

A single-laser provides cost savings, but introduces additional assignment constraints in spectrum allocations. Such constraints may be particularly relevant in next-generation metro networks, where significant traffic asymmetry is expected. For example, huge amount of traffic flows from a content delivery network (CDN) network to the access (i.e., downstream traffic to end-users) while the reverse direction (i.e., upstream traffic from users) is significantly less utilized.

Figure 4.3 shows two asymmetrical bidirectional connections. From nodes A to B , and C to D , a large amount of spectrum is occupied to guarantee the required bandwidth capacity. Figure 4.3a shows the case where a single-laser is shared between the TX and local oscillator at RX while Figure 4.3b depicts two independent laser sources are utilized, enabling a free allocation of each central frequency. As it can be noticed, in the first scenario, the constraint on the same CF for both TX and RX leads to high fragmentation and, in turn, to relevant wasting of spectrum resources, which is avoided when each card is equipped with two independent laser sources.

Figure 4.3: Resource allocation with different transponder technologies

4.1.2 Traffic Dynamicity

Regardless of the bidirectional tunability issue described in the previous subsection, of special interest in this work is taking advantage of bandwidth-variable transponders (BVTs) to follow traffic variations adapting rate and spectrum occupation.

Figure 4.4 shows two neighboring connections from nodes A to B , and C to D , (for simplicity, just one direction is depicted). Both connections experience time dependent bandwidth utilization. For example, the first connection serves a business district where most of the bandwidth is requested in working hours (time $t=0$) with scarce use, e.g., in the evening ($t=1$). Conversely, the second connection serves a residential area where limited bandwidth is needed during working hours but larger service requests are experienced in the evening. To this respect, different SA policies can be applied potentially considering time-dependent spectrum sharing.

In Figure 4.4a, the resource allocation is performed in a fixed manner, considering peak-hour traffic volumes. In Figure 4.4b, the allocation is performed accounting for multihour variations and not on the peak-hour values. This way, spectrum resources used by connection A to B at $t=0$ can be reused by connection C to D at $t=1$. In this case, fixed CFs are assumed. Compared to Figure 4.4b, in Figure 4.4c we consider re-tuning of CF as an additional degree of flexibility. Such re-tuning can be implemented by adopting the *push-pull defragmentation* technique presented in [54], where the automatic frequency control of the coherent RX is properly exploited to track the gradual shift deliberately applied to the TX, without affecting high-layer traffic. Such adaptation is further simplified in FONs networks given the absence of filters.

It is worth noting that, as discussed in the subsequent sections, the combination of common/independent CF assignment in bidirectional connections and such fixed/multi-hour assignment with/without central frequency adaptation may lead to remarkably different network utilization performances, with potential high impact on deployment costs and use of resources before fiber exhaustion, even for multi-period optimization where traffic only “grows”.

4.2. Case Study

Figure 4.4: Resource allocation at different time of the day/week

4.2 Case Study

In this section we present a case study to evaluate the performance of the different aforementioned transponder technologies and their impact depending on the traffic dynamicity. We devised an algorithm for SA in filterless networks and a testing scenario to evaluate its performance.

The topology, routing and spectrum assignment (TRSA) problem in filterless networks can be decomposed into three sub-problems: (i) generation of the set of fiber trees (T), (ii) association of end-to-end demands to one of the trees (R), and (iii) spectrum allocation for each of the demands into the trees.

In this case study, we focused on horseshoe topologies, containing one bidirectional fiber tree, targeting the SA problem for all the demands within said tree. Due to the nature of FONs, topology and routing sub-problems are implicitly solved for this type of topologies.

The aim of our algorithm is to guarantee the correct provisioning of all demands while minimizing spectrum utilization. The input data of the algorithm are the following:

- FON topology containing a single fiber tree ($G(N, E)$), composed of a set of nodes with BVTs and bidirectional fibers interconnecting them, each supporting up to S slices at a line rate of R Gbps/slice.
- A list containing 24 different traffic matrices ($M_{(i,j)}^t$ where (i, j) is the source-destination node pair and $t \in [0, 23]$ the time period), containing per-hour traffic volume.

Our multi-hour spectrum allocation (MH-SA) initially considers a *semi-elastic* approach [55]: each lightpath uses a fixed CF whereas the number of occupied slices is adapted according to the offered traffic on each period.

4.2.1 Algorithm Implementation

Because of the $\mathcal{NP}$ -complexity of the SA problem, we resort to a heuristic algorithm based on biased random-key genetic algorithm (BRKGA) [56], a well-known variant of the genetic algorithm (GA) meta-heuristic. The algorithm (pseudo-code shown in Algorithm 4.1) consists on several phases that are executed sequentially within a loop until a feasible solution is found or the maximum number of iterations (given as an input parameter) is reached.

- *GeneratePopulation.* Each candidate solution (also called *chromosome*) is encoded as an array, where each index (key) represents the demand identifier and its value the CF. To generate a chromosome, all demands are randomly sorted, and one-by-one we try to find a valid CF (using a first-fit approach) that ensures enough slices available for each hour of the day. To guarantee all demands can be initially serviced, we assume unlimited slice capacity (that is, an arbitrarily large *virtual spectrum*), whose rationale is further explained. The total number of chromosomes generated in this phase is given as an input parameter of the algorithm.
- *Crossover.* New chromosomes (*offspring*) are generated in this phase. Each new chromosome inherits its values from two 'parent' chromosomes: one of them is always selected among the *elite* population, whereas the other can be *elite* or not. Without loss of generality, in the first iteration, there is no *elite* population and both parents are selected at random. We iterate over each key (i.e., demand) selecting randomly one value (i.e., CF) among the two parents. The offspring chromosomes will then become part of the general population and the parents are discarded. The number of offspring generated in this phase is given as an input parameter.
- *CostComputation.* In order to test the feasibility of each candidate solution, we iterate over the 24 sets of demands. For each demand we select the appropriate CF from the chromosome and try to allocate the required number of slices. In case of spectrum clashing (overlapping), we consider the solution as infeasible. Once we have iterated over all sets of the demands, we select those slices that have never been used and we delete them from the virtual spectrum (shifting the remaining ones to the left). If the new shrunked spectrum size is lower or equal than the *real spectrum* size (input parameter), we consider the solution feasible and the algorithm ends. Each unfeasible solution is assigned a cost, equal to the percentage of blocked traffic.
- *Mutation* To avoid stagnation, in each iteration a portion of the general population is renovated, introducing newly created chromosomes (as explained in the first phase).

To reflect the effect of using different coherent transponders equipped with single or dual-lasers, we added an input parameter to the algorithm to control the bidirectional policy of the CF: (i) *different CFs* with no restriction applied, and (ii) *same CF* which forces lightpaths to share the same CF as its bidirectional counterpart.

4.2. Case Study

```
Input:  $G(V, E)$ ,  $M_{(i,j)}^t$ ,  $S, R$ 
generatePopulation
iteration = 0
while iteration  $\leq$  maxIterations do
    crossover
    costComputation
    if feasible solution is found then
        | break
    end
    select “Elite” among the offspring
    mutation
    iteration++
end
```

Algorithm 4.1: Pseudo-code for the Multi-Hour Spectrum Allocation algorithm

The former is implemented executing two independent instances of the MH-SA algorithm, one per direction of the horseshoe. In case a solution can be found for both, the scenario is feasible. Conversely, the latter is implemented in a tricky way: we concatenate traffic demands from each (i,j) in both directions in a virtual day of 48 hours. As such, we only execute an instance of the MH-SA algorithm.

4.2.2 Spectrum Allocation Approaches

To assess the quality of our algorithm, we compare the results with two boundary solutions in terms of achievable throughput. The lower bound is given by a *non-multihour, non-elastic* approach, where spectrum allocation is made for the worst case, that is, the daily maximum (peak) across all time intervals. The upper bound is given by a multi-hour aware *full-elastic with defragmentation* approach, where CF may also change, and therefore all connections are packed each time period using *push-pull defragmentation* techniques [54].

In order to model non-elastic/full-elastic, some minor modifications were added to our MH-SA algorithm. To implement the *non-elastic* option, we removed the dynamicity of the traffic information, considering only a single demand per node-pair where the offered traffic was equal to the maximum among the 24-hour time. For the *hitless full-elastic* approach, we consider all allocated demands can be packed, removing all spectrum between connections, for each single slot, but maintaining the same left-to-right order (in terms of slice indexes) of demands. Note that keeping such left-to-right order means that the variations in the CFs can be accomplished using non-disruptive slow spectrum-shifting techniques.

4.2.3 Testing Scenario

The algorithm was develop in Java, implementing public and well-documented interfaces and latter executed using the “Offline network design” tool of Net2Plan.

As a reference scenario, we selected a horseshoe topology depicted in Figure 4.5. It is composed of

- 2 metro-core edge nodes (MCENs) providing connectivity toward core networks.
- 5 access-Metro edge nodes (AMENs) as gateways for end-users (actual producers and consumers of traffic).
- 3 nodes acting as CDN

For simplicity, we consider asymmetric, bidirectional traffic demands for node pairs MCEN-MCEN, MCEN-CDN, CDN-CDN, CDN-AMEN and AMEN-AMEN. MCEN-CDN and CDN-AMEN traffic is only considered for the closest node pairs. For example, there is only MCEN-1-CDN-1 traffic but not MCEN-2-CDN-1 traffic, as well as CDN-1-AMEN-1 traffic but not CDN-1-AMEN-5 traffic.

Figure 4.5: Reference horseshoe topology

A combination of several different methods were used to generate a realistic multi-hour traffic matrix. First, a reference traffic matrix was built using a population-distance model described in [46], using scaling factors coming from analysis, trends and forecast for the incoming years [57]. Then, to recreate a multi-hour scheme, an activity factor was applied to the aforementioned traffic to simulate variance on each hour of the day [58] using a bimodal distribution to model peak and idle periods. Parameters of these distribution like width of the peak period and peak-to-idle ratio where selected at random.

Different tests were performed using two different modulations: (i) 32 GBaud PM-QPSK over 37.5 GHz, with a total bitrate of 100 Gbps ($R=16.6$ Gbps per 6.25 GHz slice) and (ii) PM-16QAM, leading to 200 Gbps over 37.5 GHz ($R=33.3$ Gbps per 6.25 GHz slice). We establish a spectrum of 2.5 THz ($S=400$ slices). We scale increasingly the total offered traffic increasingly until blocking occurs. To determine spectrum requirements SU

4.2. Case Study

(i.e., number of occupied slices) for traffic (i, j) in time period t Equation (4.1) is applied:

$$SU_{(i,j)}^t = \left\lceil \frac{M_{(i,j)}^t}{R} \right\rceil \quad (4.1)$$

We tested the algorithm MH-SA for the three SA approaches, single/dual-laser transponders and the two modulation formats. Throughput results, indicating the peak-hour carried traffic before resource exhaustion, are presented in Figure 4.6.

Figure 4.6: Maximum throughput (in Tbps)

We observed that the semi-elastic approach outperforms the fixed approach by a 23-33% in overall throughput. As expected, the algorithm performs poorly compared to the *full-elastic* scheme (with losses of 10-24% of throughput). These numbers are presented in Table 4.1. Interestingly, even though it is clear that a full-elastic approach would be preferred in all instances, we would like remark that despite the fact that push-pull techniques [54] have been fully tested and demonstrated, unfortunately are not yet commercially available.

Table 4.1: Throughput variation (percentage) when using semi-elastic allocation compared to other strategies

SA approach	Dual-laser	PM-QPSK	PM-16QAM
Fixed	Yes	23.53%	23.91%
	No	31.03%	32.76%
Full-elastic	Yes	-23.64%	-22.62%
	No	-10.59%	-9.94%

From the results, it is worth discussing the gain in terms of throughput using dual-laser transponders instead of the ones equipped with a single laser. Surprisingly, the use of the former only yields to a throughput improvement of 10-11% and, as such, may not justify investment on these devices.

Lastly, we illustrate the efficiency of our MH-SA algorithm in Figure 4.7. This picture presents a partial snapshot of the spectrum utilization during a day in the case of using

Table 4.2: Throughput gain (percentage) when using dual resonator transponders instead of single resonator

SA approach	PM-QPSK	PM-16QAM
Fixed	17.2%	18.9%
Semi-elastic	10.5%	11%
Full-elastic	29.4%	29.2%

same CF for TX/RX. In Figure 4.7a we can see the poor spectrum utilization of the fixed approach, since there is no change for spectrum sharing. Conversely, for the semi-elastic approach (see Figure 4.7b) it is clear that because of bandwidth adaptation, slices can be shared by different demands in different time intervals, thus reducing the overall spectrum requirements. However, because of using same CF for both directions in this scenario, spectrum sharing cannot be totally exploited, thus reducing the overall throughput with respect to dual-laser transponders. Finally, the full-elastic approach yields the best spectrum utilization.

4.3 Conclusions and Further Work

In this chapter, we have presented an algorithm to determine the maximum throughput which can be achieved in filterless optical networks according to different spectrum management policies. In our results, a market-ready semi-elastic approach (considering state-of-the-art BVTs) represents a reasonable solution compared to a non-commercial hitless full-elastic scenario. Besides, we have illustrate the fact that single-laser transponders may be a cost-effective alternative to dual-laser ones with limited throughput losses.

Extensions to this work may include mesh topologies, where fiber trees are also a decision variable, and/or techno-economic studies (e.g., independent tunability of TX/RX, comparison with ROADM-based solutions, and so on).

4.3. Conclusions and Further Work

Figure 4.7: Spectrum occupation for four example demands

Chapter

5

Scalability of WaveBand-based ROADMs

5.1	Photonic Integrated WaveBand ROADMs	47
5.2	WB-ROADM Design	48
5.3	Case Study	49
5.3.1	ILP Formulation	50
5.4	Conclusions and Further Work	51

In the previous chapters of Part II, we addressed network planning topics for mesh optical networks, based on ROADMs, and linear topologies, based on filterless optical devices. In this chapter, we analyze the performance of WaveBand ROADMs as another family of devices used in metropolitan area networks. Researchers of the Consorzio Nazionale Interuniversitario per le Telecomunicazioni (CNIT) (Italy) have developed a low-cost prototype of these active switching optical devices, which aim to offer a trade-off between the switching capabilities of a traditional ROADM and the cost of passive optical nodes.

5.1 Photonic Integrated WaveBand ROADMs

Next generation metro networks are expected to provide large bandwidth availability over relatively short distances. At the transmission side, a new generation of lower-cost coherent systems is rapidly emerging. At the switching side, today's generation of Liquid Crystal on Silicon (LCoS)-based ROADMs, designed for high-performance long-reach core networks, is considered too expensive to be widely deployed in the metro segment. For this reason, silicon photonic integrated switching devices are eventually gaining traction given their potential limited cost (due to their compatibility with electronic circuits fabrication process) and their suitability for metro networking scenarios of limited complexity (e.g., horseshoe or ring topologies). For example, a 2-degree ROADM implemented as photonic integrated circuit (PIC) on silicon-on-insulator (SOI) technology has been recently designed and demonstrated for coarse-WDM metro ring applications [59].

We first present the design of a 2-degree WaveBand ROADM (WB-ROADM) [60, 61] based on multi-stage filter [62] develop by researchers from CNIT, and a case study to test its performance. This WB-ROADM exploits Complementary Metal-Oxide-Semiconductor (CMOS)-compatible SOI technology providing a single drop waveband per degree, i.e., designed to drop multiple coherent transmissions not requiring per-channel filtering (see Figure 5.1). Compared to LCoS-based ROADMs, the most relevant drawback of this SOI-based solution (for metro scenarios) stands in the large transition bandwidth (TBW) of the filtering stages, which may significantly limit the overall use of spectrum resources (only shaded spectrum in the right side of Figure 5.1 should be used). Thus, in this chapter, a case study is devised, showing an efficient TBW-aware provisioning strategy to account for large TBWs.

Figure 5.1: 2-degree WB-ROADM

5.2 WB-ROADM Design

The designed WB-ROADM exploits, in the drop section (WB-Drop module in Figure 5.1), a cascade of three filters implemented in CMOS-compatible SOI technology. Single-stage filter architecture is shown in Figure 5.2.

Figure 5.2: Reference architecture of "WB-Drop" blocks.

Each filter comprises an unbalanced Mach-Zehnder interferometer (MZI) loaded with

5.3. Case Study

two micro-ring resonators (MRRs) on its branches, implementing a 5th order half-band Chebyshev filter. Each filter presents a period that is half of the bandwidth, driving the presence of multiple pass bands within the whole C band. Thus, a cascade of three filters with different Free Spectral Range (FSR) values is exploited, successfully enabling just a single waveband to be dropped over the entire C band. In particular, *OUT1* ports of the three stages are cascaded while *OUT2* ports are recombined, enabling express signals to pass through the node.

A single stage of a similar filter has been also fabricated using 100 KeV electron-beam lithography on an SOI wafer with 220 nm thick silicon layer on 3 μm thick buried oxide (BOX) layer [62].

Figure 5.3 shows a simulation of the performance for the whole node, highlighting the combination of single drop windows achieved for different tuning settings, enabling high out-of-band rejection of 30dB over the whole C-band. The -0.5 dB of the pass-band windows is 500 GHz, with negligible insertion loss at peak wavelength. Filter tunability is expected to be obtained through active heaters. This way, by tuning the filter over one FSR, the pass-band window can be centered at any wavelength within the C band.

Figure 5.3: Filter performance (500 GHz pass-band) within the whole C band

This photonic integrated WB-ROADM is expected to be significantly cheaper than LCoS-based ROADMs. However, it presents a large -0.5-to-30dB TBW in the order of around 170/196 GHz for a filter width of 500/850 GHz, respectively, unusable for add/drop/pass-through operations. Potentially, this constraint significantly limits the overall amount of available spectrum resources compared to LCoS-based ROADMs, which present TBW values in the order of just a few GHz [63]. Note that higher filter order, e.g., using additional MRRs, may be adapted to reduce TBW values. However, implementation complexity would increase without remarkably reducing the gap with LCoS devices.

5.3 Case Study

WB-ROADMs augment the routing and spectrum assignment (RSA) problem in the following aspects: (i) lightpath grouping, and (ii) TBW avoidance. First, all incoming

optical connections to a node must have a frequency within the bounds of the passband filter. Second, add and pass-through lightpaths must avoid both drop and TBW bands. These restrictions entail careful planning where traditional spectrum assignment schemes are no longer valid (e.g., first-fit). In this case study, we present an offline network planning scheme using Integer Linear Programming (ILP), considering a WB-ROADM-based network, thus affected by the aforementioned restrictions.

5.3.1 ILP Formulation

The input parameters for the ILP formulation are: a set of nodes ($\mathcal{N}$), links ($\mathcal{E}$) and optical demands ($\mathcal{D}$). Let $\mathcal{P}(d)$ and $\mathcal{C}(d)$ denote the set of candidate paths and candidate channels for demand $d \in \mathcal{D}$ respectively (we define a channel c as a contiguous set of slices in the spectrum set ($\mathcal{S}$), i.e $c \subseteq \mathcal{S}$). $\mathcal{C}_f(n)$ represents the set of candidate filter, and $\mathcal{TB}_f(n)$ the set of respective transition bands for node $n \in \mathcal{N}$. From them, we define the following data:

- γ_{cs} : Equal to 1 if channel $c \in \mathcal{C}$ uses slot $s \in \mathcal{S}$, and 0 otherwise.
- δ_{pe} : Equal to 1 if path $p \in \mathcal{P}$ traverses fiber link $e \in \mathcal{E}$, and 0 otherwise.
- γ_{fs} : Equal to 1 if filter $f \in \mathcal{F}$ uses slot $s \in \mathcal{S}$, and 0 otherwise.
- γ_{ftbs} : Equal to 1 if transition bands of filter $f \in \mathcal{F}$, and 0 otherwise.

Also, we define the following decision variables:

- $y_{pc} \in 0, 1$, equal to 1 if path $p \in \mathcal{P}$ uses channel $c \in \mathcal{C}$, and 0 otherwise.
- y_{nf} if node $n \in \mathcal{N}$ uses filter and transition band $f \in \mathcal{F}$, and 0 otherwise.

$$\max \quad \sum y_{pc} \quad (5.1)$$

subject to:

$$\sum_{p \in \mathcal{P}(d)} \sum_{c \in \mathcal{C}(d)} y_{pc} \leq 1 \quad \forall d \in \mathcal{D} \quad (5.2)$$

$$\sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{P}(d)} \sum_{c \in \mathcal{C}(d)} \gamma_{cs} \cdot \delta_{pe} \cdot y_{pc} \leq 1 \quad \forall e \in \mathcal{E}, s \in \mathcal{S} \quad (5.3)$$

$$\sum_{f \in \mathcal{C}_f(n)} y_{nf} = 1 \quad \forall n \in \mathcal{N} \quad (5.4)$$

$$y_{pc} \cdot \gamma_{cs} \leq y_{nf}|_{n=b(p)} \cdot \gamma_{fs} \quad \forall p \in \mathcal{P}, s \in \mathcal{S} \quad (5.5)$$

$$y_{pc} \cdot \gamma_{cs} \leq 1 - (y_{nf} \cdot \gamma_{fs}) \quad \forall n \in \mathcal{N}, p \in \mathcal{P}(n), s \in \mathcal{S} \quad (5.6)$$

$$y_{pc} \cdot \gamma_{cs} \leq y_{nf} \cdot \gamma_{ftbs} \quad \forall n \in \mathcal{N}, p \in \mathcal{P}(n), s \in \mathcal{S} \quad (5.7)$$

The objective function aims to maximize the carried traffic (Equation (5.1)). Constraints (Equation (5.2)) and (Equation (5.3)) ensure each lightpath is established through a single path, whereas avoiding spectrum clashing. Constraints Equation (5.4) and Equation (5.5) guarantee all nodes have a valid filter and all incoming lightpaths fall inside the pass-band of that filter, respectively. Lastly, Equation (5.6) and Equation (5.7) enforce both add and pass-through lightpaths to avoid TBW and pass-band filter of intermediate

5.4. Conclusions and Further Work

nodes along the path. Note that our ILP may try to overlap TBW bands from different nodes.

The selected reference is based on an 8-node horseshoe topology (see Figure 5.4), typically used in real metro networks. Two sets of traffic matrices were randomly generated: (i) five matrices following a uniform distribution (0,10 Gbps], and (ii) five matrices with bimodal asymmetric traffic (75% of the demands uniform (0,10 Gbps] and 25% (0,100 Gbps]). For each traffic matrix set, two filter width values were considered: 500 and 850 GHz. Finally, we consider a spectral grid of 4 THz, with 100 Gbps bidirectional transponders and 50 GHz spectral width per lightpath.

Figure 5.4: 8-node horseshoe topology.

The scenarios were evaluated using Net2Plan, and CPLEX [64] as the solver. All matrices were scaled (total normalization) until no feasible solution could be found (i.e., at least one demand could not be satisfied). Figure 5.5 shows the maximum throughput obtained with any of the matrices. Additionally, some tests considering both types of filters having a hypothetical negligible TBW (i.e., 5 GHz) are presented. Clearly, small differences in throughput can be appreciated for 500 GHz filters. The rationale behind this behavior is that filter width, which ultimately determines the number of lightpaths that fits inside, is a more stringent limiting factor than TBW. In contrast, the 850 GHz option shows significant differences between the different TBW. This is due to the fact that filter width encompasses 21.5% of the 4 THz grid, and therefore circumventing the TBW (each node “contaminates” up to 10% of the 4 THz in the worst case with no-TBW-overlap) becomes the bottleneck as traffic grows, leading to a drop in throughput.

5.4 Conclusions and Further Work

In this chapter we showed that careful planning can alleviate throughput penalties in photonic integrated WB-ROADM-based metro networks, coming from their larger transition bands. These results encourage further analysis for such technology, in order to show its viability as a low-cost alternative to traditional ROADMs, for more cost-effective

deployment and upgrading of metro networks. Besides, a cost-throughput comparison with filterless technologies is worth the effort.

Figure 5.5: Maximum throughput achieved (in Tbps)

Part III

Multilayer Dynamic Operation Using Software-Defined Networking

Chapter 6

Network Optimization using SDN Controllers

6.1	Off-platform Application for OpenDaylight	55
6.2	Communicating Net2Plan with OpenDaylight	57
6.3	Case Study	58
6.3.1	Retrieving Topology Information	59
6.3.2	Estimation of End-to-End Traffic Volumes	60
6.3.3	Real-Time Link Utilization Monitoring	61
6.3.4	Installing Forwarding Rules	61
6.3.5	Work Tracing	62
6.4	OpenDaylight and SDON	63
6.5	Conclusions and Further Work	64

In Part II we presented several proposals about network planning but, most interestingly, in Chapter 4 we also presented different techniques for real-time network optimization, which helps us to maximize the efficiency on the use of the available network resources. Aiming to demonstrate the usability of those techniques in operational networks, Part III considers the integration of planning tools with existing SDN controllers.

In this first chapter of Part III, we report the integration of the Net2Plan planning tool (see Section 2.2.3) with the popular OpenDaylight SDN controller. We show that an off-platform application on top of the controller can help network operation tasks, providing real-time re-optimization based on link utilization.

6.1 Off-platform Application for OpenDaylight

The OpenDaylight [65] platform is an open-source project which aims to provide a flexible, modular and pluggable SDN controller. Coded in Java, OpenDaylight can be installed and deployed in any machine that supports the Java Virtual Machine (JVM).

This SDN controller can be managed and configured via command line or a web-based User Interface (UI). OpenDaylight can be extended, adding new functionalities as modules with the Java OSGi framework. These modules work internally within the controller, while a northbound API, based on REpresentational State Transfer (REST), is exposed in order to provide connectivity to external applications. These applications are able to request network information to the controller, perform some optimization and then command the controller to set and distribute new rules among the configurable devices. OpenDaylight provides base network services (Figure 6.1) such as topology discovery or statistics recollection, and new ones can be added using the aforementioned framework.

The southbound interface, which allows connectivity with network devices, supports multiple protocols and standards natively (e.g., Border Gateway Protocol with Link-State extensions (BGP-LS), Path Computation Element Protocol (PCEP), OpenFlow, and so on) making vendor interoperability possible.

Despite all out-of-the-box functionalities, modules, and supported protocols, OpenDaylight (and other SDN controllers) still needs to be programmed suiting the requirements of the user. It cannot take decisions, let alone perform complete tasks like optimize or reconfigure a network without some input. Thus, an off-platform application hierarchically on top of the controller may prove critical to overcome this lack of “logic”. The integration of OpenDaylight and Net2Plan could allow the latter to step up from a ‘planning tool’ to a full real orchestration role.

Therefore, a proof-of-concept (PoC) was proposed to ascertain this hypothesis.

Figure 6.1: OpenDaylight architecture

6.2 Communicating Net2Plan with OpenDaylight

In the proposed PoC, Net2Plan is connected to OpenDaylight using its northbound REST API. This communication is bidirectional: the topology (composed of hosts, switches and links), as well as static information about link utilization, is retrieved by Net2Plan, while programming flow rules are sent to the controller (as seen in Figure 6.2). The latter, known as southbound API, converts OpenDaylight into a proxy or adaptation layer between Net2Plan and OpenFlow devices, that is, Net2Plan computes optimal routing using abstract data models which is then translated into OpenFlow commands by OpenDaylight, and viceversa, as reported in Section 6.3.5.

Figure 6.2: Architecture overview of the Net2Plan – OpenDaylight interconnection

Two options are available for northbound communication. First, as aforementioned, the use of OSGi modules, which are developed in Java, implementing specific classes and interfaces within OpenDaylight. This method allows a complete integration as part of the controller, having access to all features and capabilities. The second option is the use of a REST API, that enables communication via the exchange of HyperText Transfer Protocol (HTTP) messages. We selected this method mainly for two reasons: (i) using OSGi modules adds a layer of complexity we want to avoid in this PoC, and (ii) the connection via a standardized REST API makes backward/forward compatibility easier.

Upon starting OpenDaylight, a web-server (default port 8080) is available. This server has two purposes, first, displaying a web user interface (WebUI), accessible via the web

browser. This WebUI (see Figure 6.3) shows the network topology, flow routing tables, and statistics among other useful information. And second, allowing the exchange of HTTP messages according to the provided REST API. Responses containing information about the network topology, statistics, and so on, are received in JavaScript Object Notation (JSON) or eXtensible Markup Language (XML) format. Note that in first versions, OpenDaylight¹ provided only a “proprietary” REST API, and newer versions may provide standardized northbound interfaces based on initiatives like Transport API [66] or OpenConfig [67] using NETCONF/RESTCONF.

For this PoC, a plugin for the GUI of Net2Plan was created using the “Offline Network Design” tool Section 2.2.3.2 as a template. The newly “OpenDaylight Northbound Manager” provides a graphical interface to interact and orchestrate the controller (using the aforementioned REST API) through the new “Connection manager”.

The panel (shown in Figure 6.4) is divided into three parts. First, some text fields allow the user to input the information to allow communication and authentication with the REST API of the controller, i.e., IP, TCP port, username, and password. Second, a text area logs the message exchange between Net2Plan and OpenDaylight, for debugging purposes. And third, three buttons corresponding to the implemented functionalities. “Retrieve topology” imports the topology from OpenDaylight so it is available and can be displayed in Net2Plan. The “Estimate traffic” button is able to estimate end-to-end traffic volume and extrapolate a traffic matrix from link load statistics. The third button, “Update routing”, translates Net2Plan routes (i.e., orderly sequence of traversed links) into OpenFlow rules, which are then forwarded to OpenDaylight for installation on the devices. It is worth mentioning that in between the execution of the first two buttons (“Retrieve topology” and “Estimate traffic matrix”) and the third one (“Update routing”), the user can use the tab “Algorithm execution” to run any of the already available algorithms integrated in Net2Plan.

We devised a case study to test this PoC, further explained in the next section. In addition, specific details of the implementation are discussed.

6.3 Case Study

We conducted several experiments to test the interaction between the OpenDaylight Northbound API and Net2Plan. These experiments were performed using an OpenFlow-based network emulated with Mininet [68, 69]. Mininet is an open-source framework that enables the emulation of a large number of hosts and OpenFlow switches, allowing them to be arranged in any desired topology and taking advantage of the inbuilt virtualization capabilities of Linux operating systems.

We set up Mininet to emulate the network depicted in Figure 6.5 (based on the 7-node Spain Network, see Appendix A.1). Mininet uses Python scripts to configure the hosts and switches to be emulated, so we added an option in Net2Plan to save any native network design as a Python script according to the documentation provided by Mininet. We assume that each Net2Plan node can represent a switch or a host, being the latter

¹The first release, Hydrogen, was used for this PoC.

6.3. Case Study

Figure 6.3: OpenDaylight WebUI

used to inject/receive traffic into the network.

After the configuration file is generated, OpenDaylight should be executed, followed by Mininet (preferably in this specific order).

6.3.1 Retrieving Topology Information

In order to orchestrate OpenDaylight, the first step should be retrieving the information relative to the emulated topology. OpenDaylight is able to detect OpenFlow switches once they “register” in the controller. Then, flow rules are installed into each switch to enable Link-Layer Discovery Protocol (LLDP) and thus the controller learns how they are interconnected. However, hosts are not shown by default. The command `pingall` is executed in Mininet, generating some traffic among all hosts. Since switches do not implement any flow rule when initialized, routing works in reactive mode. OpenDaylight routing policy is based on a Dijkstra shortest-path algorithm [70] (using the number of hops as the metric), generating rules on demand and installing them into switches.

Once OpenDaylight has discovered the whole topology (see Figure 6.3), we could request this information to be sent to Net2Plan, using the “Retrieve topology” button. OpenDaylight partitions the topology information into different groups: switches (the `SwitchManager` API returns the information about these devices), links (`Topology` API) and hosts (`HostTracker` API); thus, three queries are required to retrieve the whole network. Net2plan has been engineered to gather this information, transform and integrate it

Figure 6.4: Connection manager panel

to produce a Net2Plan design. Furthermore, additional data such as the name of network interfaces or link bandwidth is also fetched and added to the design.

Since the exchange of LLDP packets do not contain geographical location, positioning is done manually using the graphical interface of Net2Plan. Figure 6.6 shows the discovered topology as depicted by Net2Plan. Lastly, bidirectional links from the Mininet emulation are represented in Net2Plan as unidirectional links in opposite directions.

6.3.2 Estimation of End-to-End Traffic Volumes

In order to properly optimize the network, an accurate estimation of traffic volume is needed. Operators have several methods to obtain such data in real time, for example using SNMP pollings to gather link traffic. OpenDaylight provides per-flow information (**Statistics API**), as well as link load, which in conjunction with a simple approach like the gravity model [71, 72] can provide a simple estimation of a traffic matrix, suitable for the goals of the PoC.

As stated by the model in Equation (6.1), traffic $T(i, j)$ from node i to node j is given by:

$$T(i, j) = T^{in}(i) \cdot \frac{T^{out}(j)}{\sum_{e \in E^{edge}} T^{out}(e)}, \quad (6.1)$$

where $T^{in}(i)$ is the traffic from node i entering the network via the host-to-switch link, $T^{out}(j)$ is the traffic leaving the network at node j via the switch-to-host link, and E^{edge} is the set of host $\leftrightarrow$ switch links. It is important to remark that the OpenDaylight **Statistics API** returns absolute traffic volumes rather than the difference since last observation, so this computation has to be done manually to get an accurate up-to-date traffic matrix. Lastly, we remark that Net2Plan is able to implement different or more complex matrix traffic estimations. In this case study, the **iperf** tool was used to generate traffic between the emulated hosts.

Figure 6.5: Emulated topology

6.3.3 Real-Time Link Utilization Monitoring

To further exploit the `Statistics` service, an additional feature was included in the Net2Plan “OpenDaylight Northbound Manager” tool. This feature aims to monitor link utilization in real-time by making periodic polls (e.g., 10 seconds intervals). This feature can be accessed by right-clicking over any link and executing the option called “View real-time traffic”.

Since OpenDaylight is unable to provide single link information, Net2Plan is forced to gather all the information from origin/destination nodes and filter the data to get specific values for a particular link.

6.3.4 Installing Forwarding Rules

As aforementioned, in the absence of rules, OpenDaylight configures shortest-path routing. The goal of this case study is to optimize these rules, using Net2Plan to achieve minimum congestion routing. The rules are obtained solving the following ILP:

$$\min \quad \rho \tag{6.2}$$

subject to:

$$\sum_{e \in \delta^+(n)} x_{de} - \sum_{e \in \delta^-(n)} x_{de} = \begin{cases} 1, & \text{if } n = a(d) \\ -1, & \text{if } n = b(d) \\ 0, & \text{otherwise} \end{cases} \quad \forall n \in N, d \in D \tag{6.3}$$

$$\sum_{d \in D} h_d \cdot x_{de} \leq \rho \cdot u_e \quad \forall e \in E \tag{6.4}$$

Figure 6.6: Emulated topology retrieved and shown in Net2Plan

$$x_{de} \in \{0, 1\} \quad \forall d \in D, e \in E \quad (6.5)$$

$$\rho \leq 1 \quad (6.6)$$

Equations (6.2) to (6.6) aim to solve the minimum congestion capacitated multi-commodity flow problem. Notation follows a flow-link formulation [31], where given a set of nodes (N), links (E) and demands (D), ρ is the maximum utilization across all links, the variables x_{de} represent the traffic from demand $d \in D$ traversing link $e \in E$, h_d is the amount of traffic offered by demand d and u_e the capacity of link e .

Equation (6.3) are the flow conservation constraints, which dictates that for each demand, the volume of injected traffic by the ingress node, must equal the traffic that exits through the egress node. Equation (6.4) is the link utilization constraint, i.e., carried traffic by a link cannot exceed the congestion threshold. Equation (6.5) is the binary constraint, and Equation (6.6) ensures that link utilization will remain equal or less than 1 (or 100%).

In Net2Plan, routes are defined as orderly sequences of traversed links; thus, each sequence has to be converted into a set of OpenFlow rules, being each rule defined by an origin/destination IP address and an in/out port. Once a solution is found, each set of rules is installed into each switch by OpenDaylight, after being transmitted using the FlowProgrammer API.

6.3.5 Work Tracing

Lastly, the packet-analyzer Wireshark is used to capture the communication between Net2Plan and OpenDaylight during the experiment. As can be seen in Figure 6.7, all

6.4. OpenDaylight and SDON

interaction with the controller Northbound interface is based on REST API, whereas interactions with the emulated switches (Southbound) is made using OpenFlow messages.

Source	Destination	Protocol	Info
127.0.0.1	127.0.0.1	HTTP	GET /controller/nb/v2/switchmanager/default/nodes HTTP/1.1
127.0.0.1	127.0.0.1	HTTP	HTTP/1.0 200 OK
Topology retrieval	127.0.0.1	HTTP	GET /controller/nb/v2/topology/default HTTP/1.1
			HTTP/1.0 200 OK
Link load retrieval	127.0.0.1	HTTP	GET /controller/nb/v2/hosttracker/default/hosts/active HTTP/1.1
			HTTP/1.0 200 OK
Link load retrieval	127.0.0.1	HTTP	GET /controller/nb/v2/statistics/default/port/node/OF/00:00:00:00:00:00:01 HTTP/1.1
			HTTP/1.0 200 OK
Flow update	127.0.0.1	HTTP	GET /controller/nb/v2/flowprogrammer/default HTTP/1.1
			HTTP/1.0 200 OK
Flow update	127.0.0.1	HTTP	DELETE /controller/nb/v2/flowprogrammer/default/node/OF/00:00:00:00:00:00:01/staticFlow/flow1 HTTP/1.1
			HTTP/1.0 200 OK
Flow update	127.0.0.1	HTTP	Type: OFPT_FLOW_MOD
			PUT /controller/nb/v2/flowprogrammer/default/node/OF/00:00:00:00:00:00:01/staticFlow/flow1 HTTP/1.1
Flow update	127.0.0.1	HTTP	HTTP/1.0 200 OK
			Type: OFPT_FLOW_MOD

Figure 6.7: Traces showing the message exchange for OpenDaylight

6.4 OpenDaylight and SDON

The goal of this chapter was only to describe a PoC for integration between Net2Plan and OpenDaylight to optimize an OpenFlow network. However, OpenDaylight has been reported to be used as a SDN controller for optical networks, which is referred to as software-defined optical networking (SDON).

Compared to regular SDN, SDON enables control down to the photonic level operation of optical communications, allowing for flexible adaptation of the photonic components supporting optical networking functionalities [73]. In other words, control of the optical signal transmission characteristics, like modulation format, and path, as well as monitoring optical signal quality, which can be used to dynamically control the transceivers and/or filters in switching elements.

For some time, there was an interest in augmenting OpenFlow to support optical extensions [74, 73]. However, initiatives such as OpenROADM [75] or Transport API [66] led to NETCONF-based solutions, much more aligned with vendor implementations. Therefore, SDN controllers, and not only OpenDaylight, support Network Configuration Protocol (NETCONF)/Yet Another Next Generation (YANG) capabilities based on those initiatives.

For OpenDaylight, one of the most prominent applications is “TransportPCE” [76]. This application aims to control an optical transport infrastructure using a non-proprietary south bound interface (SBI) such as OpenROADM [75]. Besides, it may be interconnected with higher-layer controllers and/or an orchestrator through a non-proprietary north bound interface (NBI). Control capabilities include, but are not limited to, the configuration of optical equipment or service provisioning, which matches with the PoCs demonstrated in both chapters of Part III. The architecture of TransportPCE is presented in Figure 6.8. More information about its components can be found in [76].

Figure 6.8: TransportPCE architecture

6.5 Conclusions and Further Work

Software-Defined Networking (SDN) has been proved to be a real paradigm shift in network management and orchestration, with the challenges and opportunities that this entails. The evolution of SDN controllers has resulted in the rise of gargantuan projects, backed not only by the academy but also by the industry with the aim to provide controllers with full interoperability.

In this chapter, we presented as a PoC a testbed for orchestration of OpenFlow-based networks using OpenDaylight as the controller and Net2Plan as the decision-maker. This integration was published in [77]. Nowadays the concept of off-platform application is widely used [78] and can even be considered the de-facto standard. Other works have considered this approach, e.g. [79], where the authors follow a similar approach and created a Net2Plan plugin [80] to integrate the planning tool with ONOS[81].

Dynamic Operation of Multilayer Optical Networks

7.1	End-user Service Deployment by Network Operators	65
7.2	Application-Based Network Operations	66
7.3	Communicating Net2Plan with ABNO	68
7.4	Multilayer Dynamic Operation	70
7.5	Case Study	71
7.5.1	Workflow 1: Establishing PCC-PCE connection	71
7.5.2	Workflow 2: Service provisioning	71
7.5.3	Workflow 3: Reaction to fiber link failure	73
7.6	Conclusions and Further Work	73

The goal of Part III is to demonstrate the integration between network planning tools and SDN controllers. As a first step, in Chapter 6 we presented a use case where Net2Plan is integrated with OpenDaylight for on-demand re-optimization of plain IP OpenFlow-based networks. Targeting SDON, where also the optical layer is operated, this chapter presents a case study using a reference implementation of the Application-Based Network Operations (ABNO) architecture from Telefónica I+D (Spain) [82, 83], based on standard protocols like PCEP and BGP-LS which are prevalent in carrier-grade commercial equipment.

7.1 End-user Service Deployment by Network Operators

The planning and operation of multilayer optical networks entail an inherent complexity. On the one hand, operators and service providers often have separate departments for the management of the IP and optical layers with little-to-none cooperation between them.

This lack of communication causes the deployment of end-user services to take several days, even weeks to be deployed and managed, as shown in Figure 7.1 (diagram adapted from [2]). In addition, network provisioning, as well as other workflows, requires human intervention, resulting in a slow and costly network operation.

Figure 7.1: Workflow for deployment of end-user services

However, in the past few years, new services (e.g., high-definition video streaming, online gaming...) have emerged, requiring not only additional bandwidth but also flexibility and dynamicity. Therefore, network operators are forced to explore solutions based on multilayer orchestration [84], which are expected to help optimize workflows in service provisioning, improve reduce OPEX and solve problems caused by the lack of interoperability between manufacturers.

The SDN paradigm in conjunction with entities such as path computation elements (PCEs) is vital to achieving a clean, automated, interoperable and fast multilayer operation.

7.2 Application-Based Network Operations

SDN has rapidly become the predominant paradigm for network orchestration, and as seen many times before, it has highlighted that clean-slate design approaches are prone to protocol and interface proliferation, which can limit deployment evolvability and device interoperability¹. As a solution, ABNO [85] is an alternative modular architecture developed with the aim to provide multi-domain and multi-technology connectivity services based on reusing existing standards and protocols.

ABNO comprises eight functional blocks, as depicted in Figure 7.2, along with their interfaces; however, production deployment does not strictly require the implementation of all components. A core component of the architecture is the *ABNO controller*. This element allows applications as well as the Network Management System (NMS)/Operational Support System (OSS) to specify end-to-end path requirements and access path state information. A path request triggers the controller to inspect current network connectivity and allocated resources and to provision a path that fulfills the resource requirements while not violating the network policy. Furthermore, the controller is also responsible for

¹As humorously explained by Randall Munroe in XKCD: <https://xkcd.com/927/>

7.2. Application-Based Network Operations

re-optimizing paths at run-time, while considering other path requests, current state, and network errors [86].

Another element of the architecture is the *Operations, Administration, and Management (OAM) handler*, responsible for collection network errors from all network layers. The Operations, Administration, and Management (OAM) handler monitors the network and collects error notification from the devices, using interfaces such as IPFIX and NETCONF, which are correlated in order to generate high-level error reports for the ABNO controller and the NMS. In addition, the ABNO architecture is integrated with the network routing policy via an *Interface to the Routing System (I2RS) client* [87]. As an IETF working group, I2RS aims to provide an architecture for real-time and event-based application interaction with the routing system of network devices. Consequently, the I2RS client of the ABNO architecture aggregates information from network routers in order to adapt its routing policy, by modifying routing tables to reflect path availability.

The *PCE controller* provides path selection, while a *provisioning manager* is in charge of path deployment and configuration, using existing control plane protocols like OpenFlow and NETCONF. It is worth noting that these functional protocols may be omitted in production deployment and the architecture proposes multiple overlapping control channels. Moreover, the architecture also contains an optional *Virtual Network Topology Manager (VNTM)*, which can provision connectivity in the network physical layer, like mapping virtual links to optical paths in WDM networks.

Topology discovery is paramount for any path selection algorithm from the PCE controller, the ABNO architecture uses multiple databases to store this relevant information. The *Traffic Engineering Database (TED)* is a requirement for any ABNO implementation and contains the network topology along with link capability and resource information. The database is populated with information collected from traffic engineering extensions in the routing protocol. Optionally, the architecture suggests support for a *LSP database*, which stores information for established paths, and a database to store the associative information between physical links and network paths, used for link capacity prediction during virtual link provision over optical technologies.

A critical element for production deployment is the capability of the ABNO architecture to apply a common policy for all path selection decisions. The ABNO architecture incorporates a *Policy Agent* which is controlled by the NMS/OSS. This policy agent authenticates requests, maintains accounting information and reflects policy restrictions for the path selection algorithm. The policy agent is a centerpiece in the architecture and any decision taken by the ABNO controller, the PCE controller and the Application-Layer Traffic Optimization (ALTO) server requires checking compliance with the active network policy. In addition to the ABNO control interfaces, the architecture provides additional application interfaces allowing to expose network state information through an ALTO server. The ALTO protocol is used to provide accurate path capacity and load information to applications and assist the application server selection process and performance monitoring.

In this chapter we present an active stateful multilayer PCE, working within the framework of the ABNO implementation of Telefónica. Communication and exchange of requests and replies with the rest of the elements of the architecture are based on

Figure 7.2: ABNO architecture

IETF standards and drafts, those related to PCEP and BGP-LS, thanks to the library `netphony-network-protocols` [82, 83], developed by Telefónica. With the stateful capability, the centralized PCE has complete multilayer information for path computation, and active means that it is able to issue recommended actions (i.e., lightpath or LSP rerouting upon failure) without any previous explicit request from the Path Computation Client (PCC). This PCE is implemented as an external plugin of Net2Plan. We demonstrate the integration within the ABNO architecture using a multilayer network emulated scenario (also developed as a Net2Plan plugin). Lastly, we present a case study based on typical multilayer workflows of network operators, including service provisioning across multiple layers and dynamic restoration under failures in the infrastructure.

7.3 Communicating Net2Plan with ABNO

The reference architecture for the integrated multilayer PCE scenario is depicted in Figure 7.3. The physical network is composed of IP/Multi-Protocol Label Switching (MPLS) routers and optical equipment. As shown in Figure 7.3, the testbed is based on the interplay of two Net2Plan² instances.

The first step is being able to maintain network state, this was achieved by implementing a basic multilayer emulator, as a new tool, using the graphical interface of Net2Plan (see Figure 7.4a). In addition, an NMS/OSS is also emulated, allowing the orchestration and management of the network, and therefore the operator is able to monitor the network state as well as establish new services, either (i) MPLS-Traffic Engineering (TE) paths (label-switched paths (LSPs)) or (ii) optical lightpaths. We assume established MPLS

²Version 0.4.0 was used for this experiment.

7.3. Communicating Net2Plan with ABNO

Figure 7.3: Architecture of the Net2Plan - ABNO interconnection

services are permanent. Additionally, this emulated controller can create fiber and nodes failure/repairs events. Similar to the OpenDaylight PoC, a plugin for the GUI of Net2Plan was developed for user interaction. More details can be found in Section 7.5.

Secondly, the external multilayer PCE is intended to receive requests from the aforementioned network emulator (acting as PCC), process them and issue the appropriate responses. Furthermore, the PCE is considered “active”, i.e., it will proactively react when a failure/repair occurs (notified by the PCC via BGP-LS updates), suggesting actions to avoid traffic loss: rerouting MPLS services, setting up new lightpaths, etc. This first prototype runs as a daemon on top of Net2Plan command line interface, as shown in Figure 7.4b.

To ensure interoperability and the autonomy of the implemented modules, communication between PCC and PCE is realized using exclusively PCEP [88] and BGP-LS [89], as reported in Section 7.5, so that any of the two parts can be transparently replaced, thanks to the usage of the Netphony suite [82, 83] which has been reported as interoperable with other state-of-the-art implementations [90]. To open the communication, the PCC establishes two sockets with the PCE. The first socket is used for bidirectional PCEP communication. On the other hand, the second one is intended for unidirectional interaction from PCC to PCE (excluding Open and Keep-Alive messages); the PCE receives BGP “Update” messages, informing about the topology or failure/repair events.

(a) Network emulator and PCEP/BGP-LS client (PCC)

(b) PCE daemon and command line interface

Figure 7.4: Screenshot of the implemented modules

7.4 Multilayer Dynamic Operation

The main challenge of active multilayer operation algorithm is to set up new LSPs in different layers taking into account the different layer-specific constraints. That means, these algorithms should be able to establish lightpaths over the physical topology, and also route connections in the virtual, lightpath-based, topology (client network). Last, but not least, they should be able to react automatically to failure/reparation events, and inform the PCC of the appropriate actions to be performed. Due to its active and stateful nature, the multilayer PCE can leverage the “delegate” PCEP field [91] to support LSP delegation from PCC to PCE, taking the latter full control over established lightpaths and MPLS flows. To be specific, the PCE is capable of suggesting the provision of new lightpaths and rerouting of existing connections without an explicit request from the PCC.

7.5 Case Study

The rest of the chapter is dedicated to describing a set of workflows, including screenshots and message exchange, of the implemented modules and the multilayer algorithm. We assume that the network operator is running both the NMS/OSS network emulator and the PCE (either on the same machine or different ones, it is irrelevant). The former can be accessed after installing the Net2Plan plugin³, then selecting the tool “Network Emulator and PCEP/BGP-LS Client” under the SDN menu (see Figure 7.4a). The latter is accessible via the command-line interface (CLI) of Net2Plan, as depicted in Figure 7.4b. For this case study, we considered a multilayer version of the NSFNet network (see Appendix A.3) where each site is assumed to have an IP router and ROADM. The packet-analyzer Wireshark was used to monitor the PCC-PCE communication using the display filter `pcep || bgp` to hide non-related packets. Lastly, we consider each fiber supports 80 WDM channels with a linerate of 40 Gbps per wavelength.

7.5.1 Workflow 1: Establishing PCC-PCE connection

To establish a connection, the PCC needs to enter the PCE IP address in the tool, and then click on the “Connect” button (see upper right side in Figure 7.4a), as defined by IETF [92] request for comments (RFC) documents. Then, an exchange of `PCEPOpen`, `PCEPKeepalive`, `BGPOpen` and `BGPKeepalive` is performed (frames 187-202 in Figure 7.5a). Once the connection is completed, the PCC sends the whole topology to the PCE. A series of `BGPUpdate` messages are sent using the Network Layer Reachability Information (NLRI) field [93] (frames 203-2015 in Figure 7.5a). Please note that several messages appear as “Malformed Packet”; at the time of this experiment, Wireshark’s packet dissector did not reflect the latest BGP specification for topology dissemination [89]. Particularly, the Netphony library implemented an early draft of the aforementioned specification⁴.

7.5.2 Workflow 2: Service provisioning

The NMS/OSS tool provides two different types of provisioning services to customers: (i) lightpath services (when “Transport layer” is selected, see upper left side in Figure 7.4a), where individual lightpaths can be established; and (ii) MPLS services (when “Client layer” is selected), where LSP deployment is performed on top of the existing virtual topology. The multilayer algorithm follows a top-down strategy, setting up new lightpaths automatically only in case not enough bandwidth is available on the client layer, and thus avoiding blocking of LSP requests. It is worth noting that for MPLS service provisioning, the connection may be established over multiple paths, in case no single one is able to carry the full request. This is possible due to the `LOAD_BALANCING` (LB) object in the `PCEPRequest` packet. The workflow consists of the following steps (frames 276-280 in Figure 7.5a)

³The source code is available at <https://github.com/jjpedreno/PCE-and-PCC-Net2Plan-Plugin>

⁴BGP Link-State Information Distribution Implementation Report, available at <https://tools.ietf.org/id/draft-ietf-idr-ls-distribution-impl-04.txt>

1. The PCC requests a new MPLS service using the “Service provisioning” option (see middle section on the right side of Figure 7.4a), then an PCEPRequest is issued (see Figure 7.5b). Bandwidth can be requested in a range from 10 to 100 Gbps (in this experiment, 100 Gbps was selected). Optionally, a LOAD_BALANCING can be included, relaxing the routing policy to serve the LSP over several paths (as shown in Figure 7.5b).
2. Upon receiving the request, the PCE tries to find a sequence of lightpaths (or a set of them if LB is enabled) over the virtual topology, using a capacitated shortest path. In case no links are available, the PCE will try to set the necessary end-to-end lightpaths to carry the requested connection. The algorithm uses a shortest path routing with first-fit for the wavelength assignment.
3. If no path (or more than one in case of LB) could be found, the PCE returns a NO_PATH reply, and the request is considered withdrawn. Otherwise, the next step is performed.
4. The PCE issues a PCEPReply message (see Figure 7.5c) including the path (or paths) and wavelength found. In case LB is enabled and multiple paths need to be returned, the PCEPReply will include multiple Explicit Route Objects (EROs). If new lightpaths need to be established, this information is included in the ERO via SERVER_LAYER objects.
5. The PCC then updates the network state and issues a PCEPReport message with the delegate field set to 1. Thus, the PCE has from now on control over the established lightpaths and MPLS services, i.e., failures will be proactively handled by the PCE without waiting for an explicit request from the PCC.

Figure 7.5: Traces from Wireshark for the ABNO - Net2Plan experiment

7.6. Conclusions and Further Work

7.5.3 Workflow 3: Reaction to fiber link failure

From the emulated NMS/OSS, the PCC can emulate a fiber link failure in order to evaluate the reaction of our multilayer PCE algorithm, using the “Simulate failure” option (see the lower right side in Figure 7.4a). We would like to remark that the PCC sends topology updates via BGP-LS in case of fiber failure, and since all provisioning has been “delegated” to the PCE (as explained in Section 7.5.2), the latter can react without an explicit `PCEPRequest` message. The multilayer restoration algorithm follows a bottom-up strategy, i.e., rerouting failed lightpaths, and if this is not possible, restoring affected MPLS services over the existing virtual topology. The workflow is as follows (frames 344-346 in Figure 7.5a):

1. The PCC informs of the failure via a `BGPUpdate` message with the `MP_UNREACH` attribute, pertaining to the failed fiber, as indicated by the NLRI.
2. The PCE tries to reroute all failed lightpaths over the surviving physical topology. In case some lightpath could not be rerouted, the PCE then tries to restore MPLS services by looking for a sequence of lightpaths (or a set of them) with enough capacity to carry the affected connections.
3. The PCE then sends back to the PCC a list of ordered actions to be taken as a `PCEPUpdate` message (see Figure 7.5d) (with the `delegate` field set to 1).
4. The PCC acknowledges the received message by issuing a `PCEPReport`, again with the `delegate` field active.

Figure 7.6 shows the effect of lightpath rerouting upon fiber failure. In this example, a failure in the fiber between nodes 2 and 5 (red dashed line) triggers the rerouting of the optical path (original and rerouted fiber paths are highlighted with dashed and solid blue lines, respectively).

Finally, in our case study we do not consider any “reverting back to normal” mechanism, where lightpaths rerouted during failure states are reverted to their original state upon reparation, in order to avoid traffic outage during lightpath migration [94]. We can actually simulate the reparation of the failed fiber link using the “Simulate reparation” option, which sends a BGP-LS reparation event (`BGPUpdate` with the `MP_REACH` attribute), but no further action will be performed by the PCE.

7.6 Conclusions and Further Work

In this chapter, we demonstrated the capabilities of an active stateful multilayer PCE with full topological information, based on the open-source Net2Plan tool, able to inter-communicate with an ABNO architecture using PCEP and BGP-LS protocols. Thanks to the usage of standard communication protocols, we should be able to substitute our emulator with real equipment, considering that such implementation from Telefónica has been already validated by third-parties [90]. Furthermore, great efforts are being applied to bring the advantages of SDN to optical devices, projects like Transport API [66] aim to provide full end-to-end control over disaggregated optical transport networks [95].

On the other hand, platforms like ONOS or OpenDaylight have proven that SDN

Figure 7.6: Lightpath rerouting upon fiber failure

controllers have evolved beyond providing a common interface to program network devices; within an ABNO architecture, these controllers can take the role of not only PCE, but also TED/LSP database, VNTM, Provisioning Manager and so on. Therefore, even though ABNO continues to be the reference architecture in many deployments, a vast majority of the functionality is provided by an SDN platform, with its own internal architecture and modules. Further work with Net2Plan should consider a more thorough integration as was explored in Chapter 6. In the end, regardless of the architecture and controller used, the focus should be put on maintaining active stateful multilayer capabilities, as well as introducing proactive reoptimization in order to fulfill the concept of *in-operation network planning* [33].

Part IV

Dynamic VNF Placement and Operation on 5G Metro-Access Networks

Service Chaining in Mobility 5G Scenarios

8.1	The Dawn of the 5G Era	77
8.2	Joint Radio-Cloud Architecture	78
8.3	Joint Orchestration Case Study	80
8.3.1	Non-orchestrated Scenario	82
8.3.2	Orchestrated Scenario	85
8.4	NFV Handover Case Study	86
8.5	Conclusions and Further Work	88

As a contribution in Part IV, in this chapter we present different techniques for VNF placement in 5G edge networks. Particularly, we consider dynamic scenarios, where a small area is densely populated and the offered traffic and service requests change drastically. We present two different case studies. On the one hand, we show that a joint orchestration of the radio and NFV domains may mitigate user blockage due to lack of Information Technology (IT) resources and/or bandwidth. On the other hand, we explore the benefits of NFV-handover, where coordinated VNFs (re-)allocation is necessary to avoid service disruption or even interruption in urban mobility scenarios.

8.1 The Dawn of the 5G Era

5G promises a ubiquitous solution featuring aspects like extraordinarily high speeds and capacity, multi-tenancy, fixed and wireless access network convergence, unconventional resource virtualization, on-demand service-oriented resource allocation, and automated management [96]. It calls for a fundamental change in the telecommunication infrastructures, shifting them from only data transport media to intelligence entities equipped with IT assets, i.e. compute and storage resources.

European 5G-PPP [97] has set a direction, aiming to significantly improve the technology level towards higher throughput and flexibility. Efforts done at the 5G-PPP, so

far, have led to solutions able to preliminary demonstrate some 5G key performance indicator (KPI) such as higher radio coverage, more bandwidth per mobile device, increased number of connected devices, logical network slicing, lower latency, and cost/energy savings.

SDN and NFV technologies [98, 99] are two important enablers in realization of the 5G-PPP vision. With the help of NFV and SDN, the embedded IT resources at the network are employed to offer added value services, e.g. innovative media services, aiming to improve the end user's QoE and creating new business opportunities for service providers. With a closer look at the added value services, it is possible to infer that they are complex operations composed by one or a set of software applications, e.g. VNF, cooperating together towards a common goal. Depending on factors like: expected performance, the required resources to run properly, maximum number of concurrent hits, etc. in an end-to-end service offering scenario, applications forming an added value service might be placed at different networking domain, e.g. the network edge, backhaul or in a remote data center far from the end user. With the help of resource orchestration techniques [24], it is possible to coordinate interaction among each piece and deliver the desired added value service.

There are several studies available in the literature focusing on the placement and application coordination problem from a static perspective [100, 101, 102]. In particular, [100] modeled the resource allocation problem with a mixed-integer program and proposed a solution to incorporate limited physical resources into the NFV resource allocation. Authors in [101] studied the resource allocation problem, taking into account the scalability issue, which might be caused by hosting multiple VNFs over the same hardware. Lastly, in [102] authors in a static scenario studied the impact of latency requirements on the placement of VNFs constituting an added value service.

Despite the importance of the previous studies, given the targeted 5G use cases, there is no surprise to see that there is a great need to investigate dynamic user behavior. For instance, 5G-PPP suggests use cases where 5G capabilities are offered in sports stadiums or shopping malls. In this case, the traffic profile changes drastically over time as crowded events occur sporadically. This definitely demands an effort to propose solutions able to cope with the traffic changes on the fly. Actually, this topic has been covered separately to some extent in the cloud and the radio world, and the industry has also made efforts to create “smart stadiums” [103]. NFV management related actions are listed under solutions for scaling up/down resources, application migrates from one point of presence to another, etc. On the network management side, allocating more/fewer frequency bands and/or wavelengths, alternative path allocation, etc. are typical solutions.

8.2 Joint Radio-Cloud Architecture

With the imminent arrival of 5G, all segments of optical networks (transport, metro and access) must not only be upgraded but also undergo radical changes in their architecture to meet all new aforementioned expectations. To accommodate these changes, several new paradigms and architectures have been recently proposed [104, 105], all of them based on bringing the computing capabilities closer to the user, i.e., edge computing. This design

8.2. Joint Radio-Cloud Architecture

paradigm provides a number of advantages, such as reduced latency or easing the traffic load on backbone networks.

One of the proposed solutions for the access segment is called “Multi-Tier Cloud-enabled Radio Access Networks” (see Figure 8.1). This architecture comprises not only by traditional macro-cells but also micro and pico/nano ones, providing additional coverage where needed. All these cells are assumed to be cloud-enabled, that is, they are composed of IT hardware on top of the radio resources, effectively placing intelligence at the edge of the network. This type of architecture allows the leverage of the NFV paradigm, allowing to instantiate VNFs such as Network Address Table (NAT), Firewall, or other virtualized network appliances.

Figure 8.1: Multi-Tier Cloud-RAN Architecture

Traditionally, radio and cloud (NFV) domains have been independently managed, each one having its control plane, however, this joint architecture poses new challenges concerning orchestration and management. Particularly, guaranteeing QoS on dynamic scenarios where the traffic volume and performance expectations change over time is of the utmost importance. In these situations, the logical cloud-enabled RAN orchestrator needs to take into account not only the radio status (i.e., traffic volume and its geographical distribution) but also cloud capabilities (i.e., IT resources, inter-VM communication, etc.) at any moment of the service life-cycle. As a consequence, a feedback QoS ensuring loop (see Figure 8.2) is absolutely necessary. Such loop comprises three phases:

- **Monitoring.** In this phase performance parameters are collected from the different elements of the radio and cloud domain (e.g., VNFs instances and service chain status) and handed over to the next step. The exact parameters to be monitored depend on different factors, such as QoS requirements, available SLA etc.
- **Decision-making.** Once all metrics have been collected, they are to be processed. A dedicated algorithm takes into account the available IT and radio resources and a decision is taken to ensure QoS. The algorithm may range from a simple greedy heuristic to advanced artificial intelligence (AI) techniques such as machine learning.
- **Reaction.** Upon making a decision, the orchestrator needs to coordinate the underlying elements to realize and deploy the necessary resources (e.g., new VNFs and/or

service chains). Then, a new cycle of the loop begins.

Figure 8.2: Feedback QoS loop

In this chapter, we present two case studies to evaluate a joint orchestration (radio and cloud) in dynamic scenarios, where the number of requested services change drastically in a small amount of time, and therefore static planning becomes inadequate.

8.3 Joint Orchestration Case Study

To test the proposed NFV-network management we propose a case study involving an event, i.e., football match, during which traffic and performance requirements change drastically due to the amount of people present in a relatively small area. We considered an area of approximately 1.5km radius around the Camp Nou stadium in Barcelona; this area contains a real radio access network (RAN) comprising 56 traditional cellular macro-cells (geographic cell position is obtained from [106]). Inside the stadium premises, we assume the presence of 200 micro-cells to provide coverage to all the people in the seating area, as represented in Figure 8.3. Each macro-cell is directly connected to a Central Office (CO) via a 10 Gbps optical link, while all micro-cells inside the stadium are connected via a 1 Gbps optical link to a central gateway which is connected to the CO with a 10 Gbps optical link; lastly, all links are considered bidirectional. The distribution of the macro-cell antennas is depicted in Figure 8.4. We assume all sites (macro-cells, micro-cells and CO) are NFV-enabled and therefore capable of instantiating VNFs; also, the CO hosts the Evolved Packet Core (EPC) and management logic.

We devised a set of simulations using the discrete simulator tool of Net2Plan. All simulations modeled 99,354 persons (the stadium capacity), originating from specific source points around the edge of the scenario (see Figure 8.4), represented by metro/bus stations, parking lots, and so on. Each person travels at a random walking speed towards the stadium, and once everyone is inside, the football match begins. For simplicity, we established the match duration to be 5 minutes, and during this time, all people remain seated and stationary inside the seating area. Once the match has ended, each person travels back to its original source point at a random walking speed, and the simulation ends when everyone has reached its destination.

Throughout all the simulation time frame, at every time slot, each person may request

8.3. Joint Orchestration Case Study

Figure 8.3: Distribution of micro-cells throughout the stadium seating area

a service among the following types: *5G Service Bundle* (i.e., web browsing, 4K video streaming, augmented reality (AR)), *Video Conference* and *VoIP*. Each service type has a minimum bandwidth requirement and associated chained VNFs [101], as outlined in Table 8.1. The probability for a user to request a service at a given time slot is based on a Binomial distribution where the parameter p is computed using the formula in Equation (8.1):

$$p = \frac{H_d}{n \cdot \sum_{s \in S} p_s \cdot B_s} \quad (8.1)$$

where H_d is the total average offered traffic at any given time and n the number of persons in the simulation. S represents the set of different service types (*VoIP*, *Video Conference* and *5G Bundle Service*), while p_s and B_s represent the probability of each request to be from a different type, and the minimum bandwidth requirement for said service, respectively.

Table 8.1: Service type bandwidth requirement and associated service chain

<i>Service</i>	<i>Chained VNFs</i>	<i>Bandwidth requirement</i>
VoIP	NAT-FW-TM-FW-NAT	250 Kbps
Video Conference	NAT-FW-TM-VOC-IDPS	2 Mbps
5G Service Bundle	NAT-FW-TM-WOC-IDPS	4 Mbps

* IDPS: Intrusion Detection Prevention, FW: Firewall, NAT: Network Address Translation, TM: Traffic Monitor, VOC: Video Optimization Controller, WOC: WAN Optimization Controller

Lastly, for all simulations, we establish the following parameters: 10 seconds per time slot, 15 Gbps average total offered traffic, each service requested has a probability of 50% to be *5G Service Bundle*, 20% for *Video Conference* and 30% for *VoIP*. All service types have a common holding time of 100 seconds. Each macro-cell site is assumed to have a server with 16 CPU cores, 64 GB of RAM and 10 TB of HDD. Micro-cell sites have a server with 8 CPU cores, 32 GB of RAM and 7 TB of HDD. The CO contains 100 CPU cores, 480 GB of RAM and 27 TB of HDD. The hardware requirements to instantiate each VNF and their concurrent number of users per instance (i.e., the amount of simultaneous services being handled) is shown in Table 8.2 [107].

Figure 8.4: Macro-cell distribution around the Camp Nou Stadium (Barcelona)

Table 8.2: Number of concurrent users per instance and hardware requirements per VNF

VNF	# of concurrent users per instance	Hardware requirements
IDPS	2500	CPU: 2 cores, RAM: 2GB, HDD: 10GB
FW	2500	CPU: 2 cores, RAM: 3GB, HDD: 5GB
NAT	3000	CPU: 1 core, RAM: 1GB, HDD: 2GB
TM	2500	CPU: 1 core, RAM: 3GB, HDD: 2GB
VOC	1000	CPU: 2 cores, RAM: 2GB, HDD: 20GB
WOC	1500	CPU: 1 core, RAM: 2GB, HDD: 10GB

8.3.1 Non-orchestrated Scenario

Given the aforementioned assumptions and parameters, the first simulation is executed. Each time a service is requested it is handled by the nearest cell [108] and the pseudo-code for the allocation algorithm is presented in Algorithm 8.1

In summary, first the algorithm checks if the handling cell has enough link bandwidth with the CO to carry the service offered traffic, blocking the request otherwise. Then, it tries to find available VNFs to traverse the chain dictated by Table 8.1. If a particular VNF from the chain is not instantiated or lacks enough capacity, a new instance will be allocated. In case the cell has not enough hardware resources to deploy new instances, the rest of the VNFs of the chain will be deployed at the CO, while always trying to use the ones already instantiated. We remark that in order to process a request, the VNFs have to be traversed in strict order from the handling cell to the CO, i.e., once a VNF is deployed in the CO, the rest of the chain must be traversed in the same site

8.3. Joint Orchestration Case Study

```
foreach serviceRequest at time  $T$  do
  servingSite = Nearest cell to the user from serviceRequest

  if link to CO has NOT enough capacity to allocate traffic from serviceRequest
  then
    | block the request
    | return
  end

  foreach requiredVNF in Service Chain associated to serviceRequest do
    if requiredVNF is deployed and with enough capacity to allocate
    serviceRequest then
      | allocate serviceRequest in existing VNF instance
      | break
    end
    if enough hardware capacity to deploy new instance of requiredVNF then
      | deploy new instance of requiredVNF
      | allocate serviceRequest in existing VNF instance
      | break
    end
    if servingSite == CO then
      | block the request
      | break
    end

    servingSite = CO
    allocate serviceRequest (deploy new requiredVNF instance if necessary)
  end
end
```

Algorithm 8.1: Pseudo-code for service request allocation and Service Chain deployment

(no going backward or loops). Since the CO is over-dimensioned in terms of hardware capabilities, there is no possibility of blocking due to a lack of IT resources. Lastly, once all requests at a given time slot have been processed, idle VNFs (i.e., not handling any service) will be de-instantiated to free unused resources. This approach will be called the “non-orchestrated case”, since we assume there is no communication between the radio control plane and the NFV control plane.

Figure 8.5: Hardware utilization of the non-orchestrated scenario

The average hardware utilization (CPU, RAM and HDD) of the macro-cells, micro-cells and Central Office is shown in Figure 8.5. It can be observed that all three resources follow a similar pattern: macro-cells are used by clusters of people near the origin points of the simulation, and their utilization slowly decreases (people gradually arrive at the stadium and begin to use exclusively the micro-cells), reaching a 0% value around minute 41. At this point, everyone is inside the stadium and no macro-cells are used. Around minute 46, once the match has ended, utilization increases again as people begin their journey back to their destinations, decreasing while people exit the simulation scenario. With regard to micro-cells, we can observe they have the highest utilization during most of the simulation for all three resources. This is to be expected since the server on micro-cells has fewer hardware capabilities than their macro-cells counterparts; as people approach and sit in the stadium the number the utilization grows until the match ends, and as soon as people exit the stadium utilization reaches 0%. Lastly, hardware utilization in the CO remains almost constant, except for a small peak during the match; since micro-cells easily reach maximum utilization, all VNFs that need instantiation are instead deployed at the CO.

8.3. Joint Orchestration Case Study

However, the average blocking during the simulation is approximate of 40%, and the instant blocking is depicted in Figure 8.6. Blocking rapidly increases with a peak occurring during the football match. This can be attributed to the fact that aggregation links between micro-cells and the central gateway (1 Gbps), as well as the link between said gateway and the CO (10 Gbps), act as a bottleneck, unable to carry all the offered traffic. This bottleneck and its ensuing high blocking rate are unacceptable in terms of QoS/QoE.

Figure 8.6: Blocking in each instant of the simulation

8.3.2 Orchestrated Scenario

From the first case, we can derive that a joint orchestrated approach could help reduce service blocking. For this second simulation, a small modification to the algorithm presented in Algorithm 8.1 was made. If the nearest, and therefore, serving cell lacks enough bandwidth to carry a service, the request is not automatically blocked (as the previous case). Instead, all nearby antennas in coverage [109] (and with enough link bandwidth) are ordered (closest first). Then the algorithm will try to allocate the request on the first available cell in the list that has enough IT resources (following the same procedure explained in the non-orchestrated scenario). Blocking occurs when the service chain cannot be allocated in any cell in coverage. In both scenarios, we assume cells have enough radio capabilities to serve a request [110].

Results from the second simulation are shown in Figure 8.7. It is important to note that both scenarios present similar patterns (see Figure 8.5), with the crucial difference that in the orchestrated case, all requests that were previously blocked in micro-cells due to the lack of bandwidth are now rerouted through nearby macro-cells. However, a difference can be appreciated, macro-cell utilization decreases slowly before the football match, not reaching 0% during said match. Since all service requests that can not be handled by micro-cell are instead served by the nearest macro-cells, this explains the activity during that time window. The changes made in the allocation algorithm make possible to serve all requests, achieving 0% blocking during all the simulation (see Figure 8.6).

Figure 8.8 illustrates this by showing the average link utilization of micro and macro-

Figure 8.7: Hardware utilization of the orchestrated scenario

cells in both scenarios (orchestrated and non-orchestrated). In both situations, micro-cell utilization is the same, services that cannot be allocated are blocked in the non-orchestrated case, and there is no link utilization of the nearby macro-cells.

However, the orchestrated case reflects that all the user requests that cannot be allocated by micro-antennas, are rerouted through nearby macro-antenna raising their link utilization while achieving 0% blocking. We can also observe a peak of utilization around minute 50 (after the end of the football match) in the non-orchestrated case. Since all people leaving the stadium are trying to request services from the nearest macro-cells, this causes a peak in link utilization. Meanwhile, in the orchestrated case the utilization increase is less pronounced since macro-cells are already used during the football match, and the service requests are distributed among several antennas further from the stadium.

8.4 NFV Handover Case Study

The previous case study showed the benefits of a joint cloud-network orchestration. In this section, we explore another key use case affected by said need of orchestration: the handover functionality. Handover mechanisms have been recurrently used in RANs to handle the movement of users across multiples antennas without disrupting phone calls or data usage. In the NFV domain, a handover implies cloud resources reassignments such

8.4. NFV Handover Case Study

Figure 8.8: Average link utilization in both scenarios

as data migration or VNFs de/instantiation; however, to the best of our knowledge, no scheme has been yet devised to cover handover actions coordinating radio, network and cloud resources.

Let us suppose a user is actively using a service requiring a specific service chain (Figure 8.9a), which is partially or totally deployed at the cell serving said user. The user then moves away from the original antenna and under the coverage of another one. The radio handover mechanism takes care of not disrupting the connection with the CO, but it may be the case that the new cell has not yet deployed the necessary VNFs to provide the service chain. Deploying new VNFs or migrating the ones from the previous cell, which we refer to as “hard handover”, takes some time (e.g., tens of seconds) and could result in degradation or even interruption of the service (Figure 8.9b).

We proposed a “soft handover” mechanism as a reference benchmark, being this a conservative approach for implementing a non-disruptive NFV service on the network edge. This mechanism works as follows: as soon as a user with an active connection enters the coverage of another cell (Figure 8.9c), which is not still managing the connection, the control plane checks whether such cell has deployed the necessary VNFs to support the service in case a real handover is required, proceeding to preemptively instantiate new ones if necessary. These resources are pre-allocated until the handover occurs, or they are released in case the user leaves the coverage area without triggering such handover. Protocol-level details behind this procedure are out of the scope of this paper but would include the refinement of S1-/X2-based mechanisms [111].

To test our proposed NFV handover scheme we devised a new set of simulations based on the scenario previously discussed. The area around the Camp Nou stadium, movement of people, placement of macro and micro-cells, service types, VNFs and services chains, allocation algorithm and so on are the same as the ones on Sections 8.3.1 and 8.3.2.

In the first simulation, no dedicated mechanism for NFV handover is assumed, thus hard handover applies; i.e., each user requesting a service is handled by the nearest cell which will try to use an existing service chain or deploy new VNFs if needed. In case the

Figure 8.9: Service chain handover problem

cell has not enough hardware capacity, the rest of the service chain will be allocated at the CO. When the user has actively moved away from the coverage of the serving cell and onto the coverage of another, and thus the service is handled to the new cell, the same process is repeated: the service is allocated into an existing service chain, or new VNFs are instantiated when needed.

On the other hand, the second simulation implements the soft handover mechanism for proper NFV orchestration. When a user with an active service is under the coverage of another cell (even if the latter is not handling the service), said cell will ensure that enough resources are allocated and available to carry the service in case it is handled to it, preemptively deploying VNFs if needed. The average hardware utilization (i.e., CPU, RAM and HDD) of all macro-cells throughout the simulation is shown in Figure 8.10, for both aforementioned strategies. Given that a handover mechanism implies the movement of persons, the utilization of micro-cells is omitted since they are used when people are seated and stationary inside the stadium.

As expected, using an NFV handover scheme causes much higher utilization due to the pre-allocation of resources, even with no assurance that they will be used. This increase of utilization (up to 3 times), in any case reaches 100%; although at first glance it may unacceptable, we can consider it a compromise considering the worst case no-handover scenario.

8.5 Conclusions and Further Work

The emerging 5G networks and mobile edge computing, leveraging on SDN and NFV technologies, are promising solutions able to fulfill the QoS requirements of diverse internet of things (IoT) verticals (e.g., low latency and huge transmission capacity). Effective orchestration and radio status aware resource allocation is an essential element to form

8.5. Conclusions and Further Work

Figure 8.10: Comparison of average hardware utilization in NFV handover scenario

an efficient ecosystem able to address 5G requirements. Such an orchestration needs to be able to coordinate all heterogeneous elements, resources, and services in such a complex and sensitive multi-vendor environment.

In this chapter we presented two case studies; the first one shows how careful resource orchestration of 5G NFV-enable radio access networks, taking into account not only radio but also bandwidth and NFV requirements, can improve the overall 5G network performance. Such joint resource coordination allows for a better allocation of resources, improving performance and QoE on 5G networks, while reducing blocking and connection failures. This work was later continued and expanded in collaboration with researchers from Universidad de Valladolid; in [112, 113] we presented new mechanisms for orchestration of VNF deployment in cloud/Mobile Edge Computing (MEC) RANs.

The second case study shows the potential benefits of a soft-handover mechanism for NFV, much needed when orchestrating access networks enabling the MEC paradigm. Despite the promising results, there is a motivation for further research in handover mechanisms that guarantee a smooth transition between cells, taking into account instantiation times while at the same time making a more efficient resource reservation. In fact, due to the trend to integrate cloud computing in the 3GPP-LTE mobile network architecture, authors in [114] present an environment to provide the migration of resources so that the cloud can “follow” the user while moving.

Further studies will analyze other scenarios where the number of cells instantiating

the service is limited (i.e., using a sort of ACTIVE SET mechanisms as in 3G/Universal Mobile Telecommunications System (UMTS)). As an ongoing work, trade-off solutions with limited number of connected cells per user, exploiting user movement predictions and leveraging a joint radio-cloud orchestration are being studied.

Part V

Closing Discussion

Final Remarks

9.1	Main Contributions	93
9.2	Open Research Areas	94
9.2.1	SDNFV and End-to-end Service Orchestration	94
9.2.2	Delving in 5G Joint Operation	94
9.3	List of Publications	95
9.3.1	Publications in International Journals	95
9.3.2	Publications in International Conferences	96

9.1 Main Contributions

Over the course of this Ph.D. thesis, the candidate has covered several aspects in the field of multilayer optical networks, from network planning to orchestration and operation. There is a section in each chapter with specific conclusions for each work, thus the next paragraphs will broadly summarize the main contributions of Parts II to IV.

Part II presented contributions related to Network Planning. In one hand, it could be argued that its relation with the title of this document is indirect; however, on the other hand, proper management and operation requires a solid foundation in terms of static network planning and survivability. This part provides works that aim to guarantee properly dimensioned networks able to withstand failures and still guarantee an adequate service (i.e., no traffic outage, guaranteeing SLA, and on).

Part III addressed the challenges posed by Multilayer Dynamic Operation Using Software-Defined Networking. The contributions included an experiment where the integration of OpenDaylight and Net2Plan allowed the orchestration of OpenFlow-based networks. Moreover, a PoC showed the possibility of dynamic operation of a multilayer network using protocols such as BGP-LS and PCEP, by developing an active and stateful PCE.

Lastly, Part IV analyzed the conundrum of Dynamic VNF Placement and Operation on 5G Metro-Access Networks. The need of a joint orchestration of radio and cloud resources was the key idea presented and discussed in this work. This joint orchestration allows an efficient deployment of service chains (SCs), providing the expected QoS/QoE even in dynamic scenarios with high mobility of users. Also, a novel SC-handover mechanism is analyzed, showing the benefits of preemptive deployment of VNFs in 5G RAN to avoid service disruption.

9.2 Open Research Areas

9.2.1 SDNFV and End-to-end Service Orchestration

Despite the trend of SDN in recent years, focusing on the development of SDN controllers and off-platform applications on top of them there is still a lot of work to be still done in the field of network operation and automation. It could be argued that the real value of SDN falls not only on the controllers but also in the rest of the ecosystem, i.e., network devices, management platforms, and applications. While research works on planning, orchestration, and reoptimization add considerable value to the field, there is still no game-changer in the form of an out-of-the-shelf tool that can be directly plugged into a network and work flawlessly with any device regardless of the vendor.

In addition, there is no doubt about the achievements made by ETSI in the standardization of management and orchestration of VNF, i.e., the MANO framework. However, there are still issues to be addressed to be in terms of reliability, flexibility, and performance. On one hand, VNFs are nowadays coexisting with legacy NFs, which leads to a particularly challenging operation. On the other hand, proper service management should require not only an efficient deployment of VNFs and consequently SCs but also provision the necessary resources for the network to efficiently carry the user traffic and achieve efficient mechanisms.

Therefore, it is impossible to discuss end-to-end service orchestration without a proper coordination between SDN and NFV management planes. Further work will explore this coordination, with the goal of achieving full end-to-end autonomous orchestration of networks and IT resources.

9.2.2 Delving in 5G Joint Operation

It is undeniable that the coming of 5G will bring drastic changes, not only in metro and access network architecture but also in service management and operation. In Part IV we discussed the benefits of coordinated RAN-NFV management, in terms of service blocking and QoE. However, in this particular field, there is yet much to be explored. The rise of new services like AR or 4K video streaming is forcing network operators to come up with new schemes and techniques for service orchestration. Very soon, a time will come when reactive processes will not be enough to provide good service quality. The proactive allocation of resources, leveraging technologies like Machine Learning (ML) will be of the

9.3. List of Publications

utmost importance.

Unfortunately, at the moment, the joint orchestration proposed in this thesis is only an idea, with no real implementation. Further work should focus on working on top of the 5G architecture, to allow proper communication and coordination between the aforementioned domains. Although recent additions from ETSI to the NFV-MANO architecture bring new features to support 5G [115], there is still much to be done to achieve a full joint orchestration of radio and cloud resources. Further work will include the proper definition of entities and modules to realize this joint-proposal, development of proactive VNF deployment schemes, and case studies involving new dynamic scenarios (e.g., massive concerts, highly transited streets and so on.)

9.3 List of Publications

The following sections are dedicated to present all publications related to the work in this Ph.D. thesis. Publications are sorted by release date in descending order.

9.3.1 Publications in International Journals

Publications in international journals are presented in table Table 9.1

Table 9.1: Publications in international journals

#	Citation	Related to
IJ1 [116]	J.-J. Pedreno-Manresa , J.-L. Izquierdo-Zaragoza, and P. Pavon-Marino, “Guaranteeing Traffic Survivability and Latency Awareness in Multilayer Network Design,” <i>Journal of Optical Communications and Networking</i> , vol. 9, no. 3, pp. B53–B63, 2017. [Online]. Available: https://doi.org/10.1364/JOCN.9.000B53	Chapter 3
IJ2 [117]	J.-J. Pedreno-Manresa , P. S. Khodashenas, M. S. Siddiqui, and P. Pavon-Marino, “On the Need of Joint Bandwidth and NFV Resource Orchestration: a Realistic 5G Access Network Use Case,” <i>IEEE Communications Letters</i> , vol. 22, no. 1, pp. 145–148, jan 2018. [Online]. Available: https://doi.org/10.1109/LCOMM.2017.2760826	Chapter 8

#	Citation	Related to
IJ3 [113]	L. Ruiz, R. Durán, I. De Miguel, P. Khodashenas, J.-J. Pedreno-Manresa <i>et al.</i> , “A Genetic Algorithm for VNF Provisioning in NFV-Enabled Cloud/MEC RAN Architectures,” <i>Applied Sciences</i> , vol. 8, no. 12, 2018. [Online]. Available: https://doi.org/10.3390/app8122614	Chapter 8

9.3.2 Publications in International Conferences

Publications in international conferences are presented in table Table 9.2. Contributions including (*) were presented by the candidate.

Table 9.2: Publications in international conferences

#	Citation	Related to
IC1 [77]	J.-L. Izquierdo-Zaragoza, A. Fernandez-Gambin, J.-J. Pedreno-Manresa , and P. Pavon-Marino, “Leveraging Net2Plan planning tool for network orchestration in OpenDaylight,” in <i>Proceedings of the 5th International Conference on Smart Communications in Network Technologies (SaCoNeT 2014)</i> , Vilanova i la Geltrú (Spain), Jun. 2014. [Online]. Available: http://dx.doi.org/10.1109/SaCoNeT.2014.6867770	Chapter 6
IC2 [118]	J.-L. Izquierdo-Zaragoza, J.-J. Pedreno-Manresa , and P. Pavon-Marino, “Net2Plan: An Integrated Open-Source Framework for Multilayer Network Planning and In-Operation Simulation,” in <i>Proceedings of the 17th International Conference on Transparent Optical Networks (ICTON 2015)</i> , Budapest (Hungary), Jul. 2015. [Online]. Available: http://dx.doi.org/10.1109/ICTON.2015.7193724	Chapters 3 to 5

9.3. List of Publications

#	Citation	Related to
IC3 [50] (*)	J.-L. Izquierdo-Zaragoza, J.-J. Pedreno-Manresa , and P. Pavon-Marino, “Maximizing IP fast rerouting coverage in survivable IP-over-WDM networks,” in <i>Proceedings of the 41st European Conference on Optical Communication (ECOC 2015)</i> , Valencia (Spain), Sep.-Oct. 2015. [Online]. Available: http://dx.doi.org/10.1109/ECOC.2015.7341809	Chapter 3
IC4 [37] (*)	J.-J. Pedreno-Manresa , J.-L. Izquierdo-Zaragoza, and P. Pavon-Marino, “Joint Fault Tolerant and Latency-Aware Design of Multilayer Optical Networks,” in <i>Proceedings of the 20th International Conference on Optical Network Design and Modeling (ONDM 2016)</i> , Cartagena (Spain), May 2016. [Online]. Available: http://dx.doi.org/10.1109/ONDM.2016.7494054	Chapter 3
IC5 [119] (*)	J.-L. Izquierdo-Zaragoza, J.-J. Pedreno-Manresa , P. Pavon-Marino, O. Gonzalez de Dios, and V. López, “Dynamic operation of an IP/MPLS-over-WDM network using an open-source active stateful BGP-LS-enabled multilayer PCE,” in <i>Proceedings of the 18th International Conference on Transparent Optical Networks (ICTON 2016)</i> , Trento (Italy), Jul. 2016. [Online]. Available: http://dx.doi.org/10.1109/ICTON.2016.7550702	Chapter 7
IC6 [120] (*)	J.-J. Pedreno-Manresa , P. S. Khodashenas, M. S. Siddiqui, and P. Pavon-Marino, “Dynamic QoS/QoE assurance in realistic NFV-enabled 5G Access Networks,” in <i>Proceedings of the 19th International Conference on Transparent Optical Networks (ICTON)</i> . IEEE, 2017. [Online]. Available: https://doi.org/10.1109/ICTON.2017.8025149	Chapter 8
IC7 [79]	P. Sköldström, Č. Rožić, and J.-J. Pedreno-Manresa , “Making powerful friends: Introducing ONOS and Net2Plan to each other,” in <i>Proceedings of the 19th International Conference on Transparent Optical Networks (ICTON)</i> , 2017. [Online]. Available: https://doi.org/10.1109/ICTON.2017.8025040	Chapter 6

#	Citation	Related to
IC8 [121] (*)	J.-J. Pedreno-Manresa , J.-L. Izquierdo-Zaragoza, F. Cugini, and P. Pavon-Marino, “On the Benefits of Elastic Spectrum Management in Multi-hour Filterless Metro Networks,” in <i>Proceedings of the 22nd International Conference on Optical Network Design and Modeling (ONDM2018)</i> , May 2018, pp. 184–189. [Online]. Available: https://doi.org/10.23919/ONDM.2018.8396128	Chapter 4
IC9 [112]	L. Ruiz, R. J. Durán, I. de Miguel, P. S. Khodashenas, J.-J. Pedreno-Manresa et al. , “Genetic Algorithm for Effective Service Mapping in the Optical Backhaul of 5G Networks,” in <i>Proceedings of 20th International Conference on Transparent Optical Networks (ICTON)</i> , Jul. 2018. [Online]. Available: https://doi.org/10.1109/ICTON.2018.8473766	Chapter 8
IC10 [122] (*)	J.-J. Pedreno-Manresa , C. Porzi, J.-L. Izquierdo-Zaragoza, F. Cugini, and P. Pavon-Marino, “Provisioning in Metro Networks Based on Photonic Integrated WaveBand ROADMs,” in <i>Proceedings of 44th European Conf. Optical Communication (ECOC)</i> , 9 2018. [Online]. Available: https://doi.org/10.1109/ECOC.2018.8535186	Chapter 5
IC11 [123] (*)	J.-J. Pedreno-Manresa , P. S. Khodashenas, J.-L. Izquierdo-Zaragoza, and P. Pavon-Marino, “Improved User Experience by Dynamic Service Handover and Deployment on 5G Network Edge,” in <i>Proceedings of the 21st International Conference of Transparent Optical Network (ICTON)</i> , 7 2019. [Online]. Available: https://doi.org/10.1109/ICTON.2019.8840228	Chapter 8

References

- [1] “Cisco Visual Networking Index: Forecast and Methodology, 2014-2019,” Cisco Systems Inc., White Paper, May 2015. [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.html
- [2] J.-P. Fernandez-Palacios, F. J. Rios, G. Cauchie, N. Lemonnier, R. Davey *et al.*, “IP and Optical Convergence: Use Cases and Technical Requirements,” Telefónica I+D, AXTEL Mexico, Bouygues Telecom, BT, China Unicom, Colt, Deutsche Telekom, KDDI, Korea Telecom, Orange and Telecom Italia, White Paper, Jan. 2014. [Online]. Available: http://www.tid.es/sites/526e527928a32d6a7400007f/assets/532c7ce328a32d4d710006bd/White_paper_IP_Optical_Convergence__1_.pdf
- [3] “The Future of Carrier SDN Networks,” [Last accessed: October 2019]. [Online]. Available: <https://www.lightreading.com/carrier-sdn/sdn-architectures/the-future-of-carrier-sdn-networks/a/d-id/715577>
- [4] “Telefónica: SDN Vendors Need to Interoperate,” [Last accessed: October 2019]. [Online]. Available: <https://www.lightreading.com/nfv/nfv-tests-and-trials/telefonica-sdn-vendors-need-to-interoperate--/d/d-id/715627>
- [5] “Lack of a universal sdn challenge,” [Last Accessed: October 2019]. [Online]. Available: <https://www.lightreading.com/carrier-sdn/sdn-architectures/lack-of-universal-sdn-a-challenge/v/d-id/715859>
- [6] “Telecommunication Networks Engineering Group (GIRTEL),” [Last accessed: October 2019]. [Online]. Available: <http://girtel.upct.es/>
- [7] “Universidad Politécnica de Cartagena (UPCT),” [Last accessed: October 2019]. [Online]. Available: <http://www.upct.es/>
- [8] P. Pavon-Marino and J.-L. Izquierdo-Zaragoza, “Net2Plan: An Open Source Network Planning Tool for Bridging the Gap between Academia and Industry,” *IEEE Network*, vol. 29, no. 5, pp. 90–96, Sep.-Oct. 2015. [Online]. Available: <http://dx.doi.org/10.1109/MNET.2015.7293311>
- [9] “Matlab - math. graphs. programming,” [Last accessed: October 2019]. [Online]. Available: <https://www.mathworks.com/products/matlab.html>

-
- [10] M. Duelli, “Heuristic Design and Provisioning of Resilient Multi-Layer Networks,” Ph.D. dissertation, Institut für Informatik, Universität Würzburg, Würzburg (Germany), 2012. [Online]. Available: <https://opus.bibliothek.uni-wuerzburg.de/frontdoor/index/index/docId/5600>
 - [11] J. M. Simmons, *Optical Network Design and Planning*, 2nd ed., ser. Optical Networks. Springer, 2014. [Online]. Available: <https://doi.org/10.1007/978-3-319-05227-4>
 - [12] J. L. Izquierdo Zaragoza, “Planning and Optimization of Multilayer Optical Networks,” Ph.D. dissertation, Telecommunications Engineering School, Jul. 2017.
 - [13] D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, “Software-defined networking: A comprehensive survey,” *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015. [Online]. Available: <http://doi.org/10.1109/JPROC.2014.2371999>
 - [14] R. Presuhn, “Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP),” IETF, RFC 3416, Dec. 2002. [Online]. Available: <http://tools.ietf.org/rfc/rfc3416.txt>
 - [15] N. Feamster and H. Balakrishnan, “Detecting BGP configuration faults with static analysis,” in *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation-Volume 2*. USENIX Association, 2005. [Online]. Available: <https://dl.acm.org/citation.cfm?id=1251207>
 - [16] H. Jamjoom, D. Williams, and U. Sharma, “Don’t call them middleboxes, call them middlepipes,” in *Proceedings of the 3rd Workshop on Hot topics in software defined networking*. ACM, 2014, pp. 19–24. [Online]. Available: <http://doi.org/10.1145/2620728.2620760>
 - [17] ONF, “Software-Defined Networking: The New Norm for Networks,” Open Network Foundation, White Paper, 2012. [Online]. Available: <https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf>
 - [18] *OpenFlow Switch Specification*, Open Networking Foundation Std. Version 1.5, Mar. 2015. [Online]. Available: <https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-switch-v1.5.1.pdf>
 - [19] S. Azodolmolky, *Software Defined Networking with Openflow*, 1st ed. Packt Publishing, 2013. [Online]. Available: <https://isbnsearch.org/isbn/9781849698726>
 - [20] J. Sherry, S. Hasan, C. Scott, A. Krishnamurthy, S. Ratnasamy, and V. Sekar, “Making middleboxes someone else’s problem: network processing as a cloud service,” *ACM SIGCOMM Computer Communication Review*, vol. 42, no. 4, pp. 13–24, 2012. [Online]. Available: <https://dx.doi.org/10.1145/2377677.2377680>
 - [21] Y. Li and M. Chen, “Software-Defined Network Function Virtualization: A Survey,” *IEEE Access*, vol. 3, pp. 2542–2553, 2015. [Online]. Available: <http://doi.org/10.1109/ACCESS.2015.2499271>

References

- [22] F. Yue, “Network functions virtualization - Everything old is new again,” F5 Networks, White Paper, 2013, [Last accessed: October 2019]. [Online]. Available: <https://www.f5.com/pdf/white-papers/service-provider-nfv-white-paper.pdf>
- [23] R. Mijumbi, J. Serrat, J. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, “Network Function Virtualization: State-of-the-Art and Research Challenges,” *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 236–262, 2016. [Online]. Available: <https://doi.org/10.1109/COMST.2015.2477041>
- [24] ETSI, “NFV Management and Orchestration: An Overview,” ETSI, Group Specification NFV-MAN 001, 2 2014, [Last accessed: October 2019]. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/NFV-MAN/001_099/001/01.01.01_60/gs_nfv-man001v010101p.pdf
- [25] ETSI, “Networkfunctions virtualisation (nfv): Terminology for main concepts in nfv,” ETSI, Group Specification 003 V1.2.1, 12 2014. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/NFV/001_099/003/01.02.01_60/gs_NFV003v010201p.pdf
- [26] J. Halpern and C. Pignataro, “Service Function Chaining (SFC) Architecture,” IETF, RFC 7665, Oct. 2015. [Online]. Available: <http://tools.ietf.org/rfc/rfc7665.txt>
- [27] H. G. Perros, *Computer Simulation Techniques: The definitive introduction!* North Carolina State University, 2009.
- [28] “Net2Plan - The open-source network planner,” [Last accessed: October 2019]. [Online]. Available: <http://www.net2plan.com>
- [29] “MatPlanWDM: An educational tool for network planning in wavelength-routing networks,” [Last accessed: October 2015]. [Online]. Available: <http://girtel.upct.es/~ppavon/matplanwdm/>
- [30] P. Pavon-Marino, R. Aparicio-Pardo, G. Moreno-Munoz, J. Garcia-Haro, and J. Veiga-Gontan, “MatPlanWDM: An educational tool for network planning in wavelength-routing networks,” in *Proceedings of the 11th International Conference on Optical Networking Design and Modeling (ONDM 2007)*, Athens (Greece), May 2007. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-72731-6_7
- [31] P. Pavon-Marino, *Optimization of Computer Networks: Modeling and Algorithms: A Hands-On Approach*. Wiley, 2016. [Online]. Available: <https://doi.org/10.1002/9781119114840>
- [32] B. Doshi and P. Harshavardhana, “Broadband network infrastructure of the future: roles of network design tools in technology deployment strategies,” *IEEE Communications Magazine*, vol. 36, no. 5, pp. 60–71, May 1998. [Online]. Available: <https://doi.org/10.1109/35.667959>
- [33] L. Velasco, A. Castro, D. King, O. Gerstel, R. Casellas, and V. Lopez, “In-Operation Network Planning,” *IEEE Communications Magazine*, vol. 52, no. 1, pp. 52–60, Jan. 2014. [Online]. Available: <http://dx.doi.org/10.1109/MCOM.2014.6710064>

-
- [34] O. Gerstel, C. Filsfil, T. Telkamp, M. Gunkel, M. Horneffer *et al.*, “Multi-Layer Capacity Planning for IP-Optical Networks,” *IEEE Communications Magazine*, vol. 52, no. 1, pp. 44–51, Jan. 2014. [Online]. Available: <http://dx.doi.org/10.1109/MCOM.2014.6710063>
- [35] J. Rak, “Resilient Routing in Communication Networks,” in *Computer Communications and Networks*, 1st ed. Springer, 2015. [Online]. Available: <https://doi.org/10.1007/978-3-319-22333-9>
- [36] O. Gerstel, C. Filsfil, and W. Wakim, “IP-Optical Interaction during Traffic Restoration,” in *Proceedings of the Optical Fiber Communication Conference and Exposition and the National Fiber Optic Engineers Conference 2013 (OFC/NFOEC 2013)*, Anaheim (United States), Mar. 2013. [Online]. Available: <http://dx.doi.org/10.1364/NFOEC.2013.NW1F.3>
- [37] **J.-J. Pedreno-Manresa**, J.-L. Izquierdo-Zaragoza, and P. Pavon-Marino, “Joint Fault Tolerant and Latency-Aware Design of Multilayer Optical Networks,” in *Proceedings of the 20th International Conference on Optical Network Design and Modeling (ONDM 2016)*, Cartagena (Spain), May 2016. [Online]. Available: <http://dx.doi.org/10.1109/ONDM.2016.7494054>
- [38] E. Palkopoulou, O. Gerstel, I. Stiakogiannakis, T. Telkamp, V. Lopez, and I. Tomkos, “Impact of IP Layer Routing Policy on Multilayer Design [Invited],” *IEEE/OSA Journal of Optical Communications and Networking*, vol. 7, no. 3, pp. A396–A402, Mar. 2015. [Online]. Available: <http://dx.doi.org/10.1364/JOCN.7.00A396>
- [39] B. Fortz and M. Thorup, “Internet Traffic Engineering by Optimizing OSPF Weights,” in *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2000)*, Tel Aviv (Israel), Mar. 2000. [Online]. Available: <http://dx.doi.org/10.1109/INFCOM.2000.832225>
- [40] O. Gerstel, “The Age of Multi-Layer Networking,” in *Proceedings of the Asia Communications and Photonics Conference 2013 (ACP 2013)*, Beijing (China), Nov. 2013. [Online]. Available: <https://www.osapublishing.org/abstract.cfm?uri=ACPC-2013-ATH2A.2>
- [41] T. A. Feo and M. G. C. Resende, “Greedy Randomized Adaptive Search Procedures,” *Journal of Global Optimization*, vol. 6, no. 2, pp. 109–133, Mar. 1995. [Online]. Available: <http://dx.doi.org/10.1007/BF01096763>
- [42] J.-P. Vasseur, M. Pickavet, and P. Demeester, *Network Recovery: Protection and Restoration, SONET-SDH, IP, and MPLS*, 1st ed., ser. The Morgan Kaufmann Series in Networking. Morgan Kaufmann, Aug. 2004. [Online]. Available: <http://www.worldcat.org/oclc/803598271>
- [43] J.-L. Izquierdo-Zaragoza and P. Pavon-Marino, “Assessing IP vs optical restoration using the open-source Net2Plan tool,” in *Proceedings of the 16th International Telecommunications Network Strategy and Planning Symposium (NETWORKS*

References

- 2014), Funchal (Portugal), Sep. 2014. [Online]. Available: <http://dx.doi.org/10.1109/NETWKS.2014.6959255>
- [44] L. Kleinrock, “The latency/bandwidth tradeoff in gigabit networks,” *IEEE Communications Magazine*, vol. 30, no. 4, pp. 36–40, Apr. 1992. [Online]. Available: <https://doi.org/10.1109/35.135787>
- [45] J. Edmonds and R. M. Karp, “Theoretical improvements in algorithmic efficiency for network flow problems,” in *Combinatorial Optimization—Eureka, You Shrink!* Springer, 2003, pp. 31–33. [Online]. Available: https://doi.org/10.1007/3-540-36478-1_4
- [46] R. S. Cahn, *Wide Area Network Design: Concepts and Tools for Optimization*, 1st ed., ser. The Morgan Kaufmann Series in Networking. Morgan Kaufmann Publishers Inc., May 1998.
- [47] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C.-N. Chuah, Y. Ganjali, and C. Diot, “Characterization of Failures in an Operational IP Backbone Network,” *IEEE/ACM Transactions on Networking*, vol. 16, no. 4, pp. 749–762, Aug. 2008. [Online]. Available: <http://dx.doi.org/10.1109/TNET.2007.902727>
- [48] R. Hülsermann, M. Gunkel, C. Meusburger, and D. A. Schupke, “Cost modeling and evaluation of capital expenditures in optical multilayer networks,” *Journal of Optical Networking*, vol. 7, no. 9, p. 814, Sep. 2008. [Online]. Available: <http://dx.doi.org/10.1364/JON.7.000814>
- [49] J. Moy, “OSPF Version 2,” IETF, RFC 2328, Apr. 1998. [Online]. Available: <http://tools.ietf.org/rfc/rfc2328.txt>
- [50] J.-L. Izquierdo-Zaragoza, **J.-J. Pedreno-Manresa**, and P. Pavon-Marino, “Maximizing IP fast rerouting coverage in survivable IP-over-WDM networks,” in *Proceedings of the 41st European Conference on Optical Communication (ECOC 2015)*, Valencia (Spain), Sep.-Oct. 2015. [Online]. Available: <http://dx.doi.org/10.1109/ECOC.2015.7341809>
- [51] C. Tremblay, É. Archambault, M. P. Bélanger, J.-P. Savoie, F. Gagnon, and D. V. Plant, “Passive Filterless Core Networks Based on Advanced Modulation and Electrical Compensation Technologies,” *Telecommunication Systems*, vol. 54, no. 2, pp. 167–181, Oct. 2013. [Online]. Available: <https://doi.org/10.1007/s11235-013-9725-y>
- [52] G. Mantelet, A. Cassidy, C. Tremblay, D. V. Plant, P. Littlewood, and M. P. Bélanger, “Establishment of dynamic lightpaths in filterless optical networks,” *Journal of Optical Communications and Networking*, vol. 5, no. 9, pp. 1057–1065, 2013. [Online]. Available: <http://doi.org/10.1364/jocn.5.001057>
- [53] C. Tremblay, P. Littlewood, M. P. Bélanger, L. Wosinska, and J. Chen, “Agile Filterless Optical Networking,” in *Proceedings of 21st International Conference on Optical Network Design and Modeling (ONDM)*, 2017. [Online]. Available: <http://doi.org/10.23919/ondm.2017.7958526>

-
- [54] F. Cugini, F. Paolucci, G. Meloni, G. Berrettini, M. Secondini *et al.*, “Push-Pull Defragmentation Without Traffic Disruption in Flexible Grid Optical Networks,” *Journal of Lightwave Technology*, vol. 31, no. 1, pp. 125–133, Jan. 2013. [Online]. Available: <http://dx.doi.org/10.1109/JLT.2012.2225600>
- [55] M. Klinkowski, M. Ruiz, L. Velasco, D. Careglio, V. Lopez, and J. Comellas, “Elastic Spectrum Allocation for Time-Varying Traffic in FlexGrid Optical Networks,” *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 1, pp. 26–38, Jan. 2013. [Online]. Available: <http://dx.doi.org/10.1109/JSAC.2013.130104>
- [56] J. F. Gonçalves and M. G. Resende, “Biased random-key genetic algorithms for combinatorial optimization,” *Journal of Heuristics*, vol. 17, no. 5, pp. 487–525, 2011. [Online]. Available: <http://doi.org/10.1007/s10732-010-9143-1>
- [57] “Cisco – The Zettabyte Era: Trends and Analysis,” [Last accessed: October 2019]. [Online]. Available: <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.htm>
- [58] J. Milbrandt, M. Menth, and S. Kopf, “Adaptive Bandwidth Allocation: Impact of Traffic Demand Models for Wide Area Networks,” in *19th International Teletraffic Congress (ITC19)*, 2005. [Online]. Available: https://www.researchgate.net/publication/215753013_Adaptive_Bandwidth_Allocation_Impact_of_Traffic_Demand_Models_for_Wide_Area_Networks/
- [59] V. Sorianello, G. De Angelis, T. Cassese, M. V. Preite, P. Velha *et al.*, “Polarization Insensitive Silicon Photonic ROADMs with Selectable Communication Direction for Radio Access Networks,” *Optics letters*, vol. 41, no. 24, pp. 5688–5691, 2016. [Online]. Available: <http://doi.org/10.1364/ol.41.005688>
- [60] K. Ishii, O. Moriwaki, H. Hasegawa, K. Sato, Y. Jinnouchi *et al.*, “Efficient roadm-ring connecting node switch architecture that utilizes waveband routing and its realization with PLC technologies,” in *Proceedings of the 35th European Conference on Optical Communication (ECOC)*, Sep. 2009. [Online]. Available: <https://ieeexplore.ieee.org/document/5287104>
- [61] B. Schrenk, F. Laudenbach, R. Lieger, T. Lorünser, P. Bakopoulos *et al.*, “Passive ROADM Flexibility in Optical Access With Spectral and Spatial Reconfigurability,” *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 12, pp. 2837–2846, Dec. 2015. [Online]. Available: <http://doi.org/10.1109/JSAC.2015.2478719>
- [62] G. Serafino, A. Malacarne, C. Porzi, F. Fresi, G. Meloni *et al.*, “Semi filter-less drop waste network demonstration with integrated SOI optical filter,” in *Proceedings of the European Conference on Optical Communication (ECOC)*, Sep. 2017. [Online]. Available: <http://doi.org/10.1109/ECOC.2017.8346157>
- [63] C. Pulikkaseril, L. A. Stewart, M. A. F. Roelens, G. W. Baxter, S. Poole, and S. Frisken, “Spectral modeling of wavelength selective switches for flexible grid optical networks,” in *Proceedings of the Conference on Lasers and Spectroscopy and the Australian International Conference on Quantum Electronics (IQEC) and the*

References

- Conference on Lasers and Electro-Optics (CLEO) Pacific Rim incorporating the Australasian Conference on Optics Optical Fibre Technology*, Aug. 2011, pp. 247–249. [Online]. Available: <http://doi.org/10.1109/IQEC-CLEO.2011.6193635>
- [64] “CPLEX Optimizer,” [Last accessed: October 2019]. [Online]. Available: <https://www.ibm.com/analytics/cplex-optimizer>
- [65] “The OpenDaylight Platform,” [Last accessed: October 2019]. [Online]. Available: <https://www.opendaylight.org/>
- [66] “ONF Open Transport API,” [Last accessed: October 2019]. [Online]. Available: <https://github.com/OpenNetworkingFoundation/TAPI>
- [67] “OpenConfig – Vendor-neutral, model-driven network management designed by users,” [Last accessed: October 2019]. [Online]. Available: <http://www.openconfig.net/>
- [68] “Mininet: An Instant Virtual Network on your Laptop,” [Last accessed: October 2019]. [Online]. Available: <http://mininet.org/>
- [69] B. Lantz, B. Heller, and N. McKeown, “A Network in a Laptop: Rapid Prototyping for Software-Defined Networks,” in *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks (HOTNETS-IX)*, Oct. 2010. [Online]. Available: <https://doi.org/10.1145/1868447.1868466>
- [70] E. W. Dijkstra, “A Note On Two Problems In Connexion With Graphs,” *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959. [Online]. Available: <https://doi.org/10.1007/BF01386390>
- [71] M. M. Rahman, S. Saha, U. Chengan, and A. S. Alfa, “IP Traffic Matrix Estimation Methods: Comparisons and Improvements,” in *Proceedings of the IEEE International Conference on Communications 2006 (ICC 2006)*, Istanbul (Turkey), Jun. 2006. [Online]. Available: <http://dx.doi.org/10.1109/ICC.2006.254710>
- [72] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg, “Fast Accurate Computation of Large-Scale IP Traffic Matrices from Link Loads,” in *Proceedings of the International Conference on Measurements and Modeling of Computer Systems 2003 (SIGMETRICS 2003)*, San Diego (United States), Jun. 2003. [Online]. Available: <http://dx.doi.org/10.1145/781027.781053>
- [73] A. S. Thyagaturu, A. Mercian, M. P. McGarry, M. Reisslein, and W. Kellerer, “Software Defined Optical Networks (SDONs): A Comprehensive Survey,” *IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, pp. 2738–2786, 2016. [Online]. Available: <http://doi.org/10.1109/comst.2016.2586999>
- [74] ONF, “Optical Transport Protocol Extensions,” ONF, WhitePaper V1.0 ONF TS-022, Mar. 2015. [Online]. Available: https://www.opennetworking.org/wp-content/uploads/2014/10/Optical_Transport_Protocol_Extensions_V1.0.pdf
- [75] “OpenROADM,” [Last accessed: October 2019]. [Online]. Available: <http://openroadm.org>

-
- [76] “OpenDaylight Documentation – Transport PCE,” [Last accessed: October 2019]. [Online]. Available: <https://docs.opendaylight.org/en/stable-fluorine/release-notes/projects/transportpce.html>
- [77] J.-L. Izquierdo-Zaragoza, A. Fernandez-Gambin, **J.-J. Pedreno-Manresa**, and P. Pavon-Marino, “Leveraging Net2Plan planning tool for network orchestration in OpenDaylight,” in *Proceedings of the 5th International Conference on Smart Communications in Network Technologies (SaCoNeT 2014)*, Vilanova i la Geltrú (Spain), Jun. 2014. [Online]. Available: <http://dx.doi.org/10.1109/SaCoNeT.2014.6867770>
- [78] D. Sanvito, D. Moro, M. Gulli, I. Filippini, A. Capone, and A. Campanella, “ONOS Intent Monitor and Reroute Service: Enabling Plug & Play Routing Logic,” in *Proceedings of the 4th IEEE Conference on Network Softwarization and Workshops (NetSoft)*. IEEE, 2018, pp. 272–276. [Online]. Available: <https://doi.org/10.1109/NETSOFT.2018.8460064>
- [79] P. Sköldström, Ć. Rožić, and **J.-J. Pedreno-Manresa**, “Making powerful friends: Introducing ONOS and Net2Plan to each other,” in *Proceedings of the 19th International Conference on Transparent Optical Networks (ICTON)*, 2017. [Online]. Available: <https://doi.org/10.1109/ICTON.2017.8025040>
- [80] “ACINO-H2020 GitHub: Application-Centric Multi-layer Network Optimization Tool,” [Last accessed: October 2019]. [Online]. Available: <https://github.com/ACINO-H2020/multilayer-optimization-tool>
- [81] “ONOS: A new carrier-grade SDN network operating system,” [Last accessed: October 2019]. [Online]. Available: <https://onosproject.org/>
- [82] “Netphony - ABNO,” [Last accessed: October 2019]. [Online]. Available: <https://github.com/telefonicaid/netphony-abno/wiki>
- [83] V. López, R. Jimenez, O. G. de Dios, L. M. Contreras, and J. F. Palacios, “Open Source Netphony suite: Enabling multi-layer network programmability,” in *Proceedings of the 21st International Conference on Optical Network Design and Modeling (ONDM)*, 2017. [Online]. Available: <https://doi.org/10.23919/ONDM.2017.7958538>
- [84] O. Gerstel and V. López, “The Need for SDN in Orchestration of IP Over Optical Multi-Vendor Networks,” in *Proceedings of the 41st European Conference on Optical Communication (ECOC 2015)*, Sep. 2015. [Online]. Available: <http://dx.doi.org/10.1109/ECOC.2015.7341833>
- [85] D. King and A. Farrel, “A PCE-Based Architecture for Application-Based Network Operations,” IETF, RFC 7491, Mar. 2015. [Online]. Available: <http://tools.ietf.org/rfc/rfc7491.txt>
- [86] C. Rotsos, D. King, A. Farshad, J. Bird, L. Fawcett *et al.*, “Network service orchestration standardization: A technology survey,” *Computer Standards & Interfaces*, vol. 54, pp. 203–215, 2017. [Online]. Available: <https://doi.org/10.1016/j.csi.2016.12.006>

References

- [87] A. Atlas, T. Nadeau, and D. Ward, “Problem Statement for the Interface to the Routing System,” IETF, RFC 7920, Jun. 2016. [Online]. Available: <http://tools.ietf.org/rfc/rfc7920.txt>
- [88] J. Vasseur and J. L. Roux, “Path Computation Element (PCE) Communication Protocol (PCEP),” IETF, RFC 5440, Mar. 2009. [Online]. Available: <http://tools.ietf.org/rfc/rfc5440.txt>
- [89] H. Gredler, J. Medved, S. Previdi, A. Farrel, and S. Ray, “North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP,” IETF, RFC 7752, Mar. 2016. [Online]. Available: <http://tools.ietf.org/rfc/rfc7752.txt>
- [90] H. Gredler, B. Rajagopalan, C. Bowers, S. Ray, and M. Bhardwaj, “BGP Link-State Information Distribution Implementation Report,” Internet Draft draft-ietf-idr-ls-distribution-impl-04, Internet Engineering Task Force, Nov. 2015. [Online]. Available: <https://tools.ietf.org/id/draft-ietf-idr-ls-distribution-impl-04.txt>
- [91] E. Crabbe, I. Minei, J. Medved, and R. Varga, “Path Computation Element Communication Protocol (PCEP) Extensions for Stateful PCE,” IETF, RFC 8231, Sep. 2017. [Online]. Available: <http://tools.ietf.org/rfc/rfc8231.txt>
- [92] “IETF - Internet Engineering Task Force,” [Last accessed: October 2019]. [Online]. Available: <https://www.ietf.org/>
- [93] T. Bates, R. Chandra, D. Katz, and Y. Rekhter, “Multiprotocol Extensions for BGP-4,” IETF, RFC 4760, Jan. 2007. [Online]. Available: <http://tools.ietf.org/rfc/rfc4760.txt>
- [94] A. Mayoral, V. Lopez, O. Gerstel, E. Palkopoulou, O. Gonzalez de Dios, and J.-P. Fernandez-Palacios, “Minimizing Resource Protection in IP Over WDM Networks: Multi-layer Shared Backup Router [Invited],” *IEEE/OSA Journal of Optical Communications and Networking*, vol. 7, no. 3, pp. A440–A446, Mar. 2015. [Online]. Available: <http://dx.doi.org/10.1364/JOCN.7.00A440>
- [95] V. López, I. Maor, K. Sethuraman, A. Mayoral, L. Ong *et al.*, “E2E transport API demonstration in hierarchical scenarios,” in *Proceedings of the Optical Fiber Communications Conference and Exhibition (OFC)*, Mar. 2017. [Online]. Available: <https://doi.org/10.1364/OFC.2017.Tu3L.4>
- [96] P. S. Khodashenas, J. Aznar, A. Legarrea, C. Ruiz, M. S. Siddiqui *et al.*, “5G Network Challenges and Realization Insights,” in *Proceedings of the 18th International Conference on Transparent Optical Networks (ICTON)*, Jul. 2016. [Online]. Available: <https://doi.org/10.1109/ICTON.2016.7550539>
- [97] “The 5G Infrastructure Public Private Partnership,” [Last accessed: October 2019]. [Online]. Available: <https://5g-ppp.eu/>
- [98] M. Chiosi, D. Clarke, P. Willis, A. Reid, J. Feger *et al.*, “Network functions virtualisation:,” An Introduction, Benefits, Enablers, Challenges & Call for Action, White Paper, 2012, [Last accessed: October 2019]. [Online]. Available: https://portal.etsi.org/NFV/NFV_White_Paper.pdf

-
- [99] R. Jain and S. Paul, "Network virtualization and software defined networking for cloud computing: A survey," *IEEE Communications Magazine*, vol. 51, no. 11, pp. 24–31, 2013. [Online]. Available: <https://doi.org/10.1109/MCOM.2013.6658648>
- [100] T. Lin, Z. Zhou, M. Tornatore, and B. Mukherjee, "Optimal Network Function Virtualization Realizing End-to-End Requests," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM)*, Dec. 2015, pp. 1–6. [Online]. Available: <https://doi.org/10.1109/GLOCOM.2015.7417447>
- [101] M. Savi, M. Tornatore, and G. Verticale, "Impact of Processing Costs on Service Chain Placement in Network Functions Virtualization," in *Proceedings of the IEEE Conference on Network Function Virtualization and Software Defined Network (NFV-SDN)*, 11 2015. [Online]. Available: <https://doi.org/10.1109/NFV-SDN.2015.7387426>
- [102] M. Savi, A. Hmaity, G. Verticale, S. Höst, and M. Tornatore, "To distribute or not to distribute? Impact of latency on Virtual Network Function distribution at the edge of FMC networks," in *Proceedings of 18th International Conference on Transparent Optical Networks (ICTON)*, 2016. [Online]. Available: <https://doi.org/10.1109/ICTON.2016.7550576>
- [103] T. Landry, "Delivering on the Promise of Smart Stadiums," JMA Wireless, White Paper, 2018, [Last accessed: October 2019]. [Online]. Available: <https://jmawireless.com/resource/smart-stadiums-white-paper/>
- [104] A. Rostami, A. Vidal, M. A. S. Santos, M. R. Raza, F. Moradi *et al.*, "An End-to-End Programmable Platform for Dynamic Service Creation in 5G Networks," in *Proceedings of the Optical Fiber Communication Conference (OFC)*. Optical Society of America, 2017, pp. Tu3L–14. [Online]. Available: <https://doi.org/10.1364/OFC.2017.Tu3L.14>
- [105] A. R. Mishra, *Fundamentals of Network Planning and Optimisation 2G/3G/4G: Evolution to 5G*. John Wiley & Sons, 2018. [Online]. Available: <https://doi.org/10.1002/9781119331797>
- [106] "Spanish Ministry of Energy, Tourism and Digital Agenda — Information About Radioelectric Installations and Radiation Exposure Levels," [Last accessed: October 2019]. [Online]. Available: <https://geoportal.minetur.gob.es/VCTEL/vcne.do>
- [107] I. Giannoulakis, P. Khodashenas, E. Jimeno, A. Albanese, M. Wilson *et al.*, "Techniques for efficient VNF Deployment with relevant VIM extensions, Evaluation framework," H2020 SESAME Project, Deliverable 4.3, 2017, [Last accessed: October 2019]. [Online]. Available: http://www.sesame-h2020-5g-ppp.eu/Portals/0/Deliverables/SESAME_Deliverable%204.3_v1.0_Final.pdf
- [108] C.-C. Lin, "Handover mechanisms in 3GPP long term evolution (LTE)," Ph.D. dissertation, University of Technology, Sydney, 2013. [Online]. Available: <http://hdl.handle.net/10453/28011>

References

- [109] T. S. Rappaport, S. Sun, R. Mayzus, H. Zhao, Y. Azar *et al.*, “Millimeter Wave Mobile Communications for 5G Cellular: It Will Work!” *IEEE Access*, vol. 1, pp. 335–349, 2013. [Online]. Available: <https://doi.org/10.1109/ACCESS.2013.2260813>
- [110] 3GPP, “Physical layer aspect for evolved Universal Terrestrial Radio Access (UTRA),” 3GPP, TechReport 25.814 Release 7, 2006, [Last accessed: October 2019]. [Online]. Available: <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=1247>
- [111] J. Prados-Garzon, O. Adamuz-Hinojosa, P. Ameigeiras, J. J. Ramos-Munoz, P. Andres-Maldonado, and J. M. Lopez-Soler, “Handover implementation in a 5G SDN-based mobile network architecture,” in *Proceedings of the 27th IEEE Annual Int. Symp. Personal, Indoor and Mobile Radio Communications (PIMRC)*, Sep. 2016. [Online]. Available: <https://doi.org/10.1109/PIMRC.2016.7794936>
- [112] L. Ruiz, R. J. Durán, I. de Miguel, P. S. Khodashenas, **J.-J. Pedreno-Manresa** *et al.*, “Genetic Algorithm for Effective Service Mapping in the Optical Backhaul of 5G Networks,” in *Proceedings of 20th International Conference on Transparent Optical Networks (ICTON)*, Jul. 2018. [Online]. Available: <https://doi.org/10.1109/ICTON.2018.8473766>
- [113] L. Ruiz, R. Durán, I. De Miguel, P. Khodashenas, **J.-J. Pedreno-Manresa** *et al.*, “A Genetic Algorithm for VNF Provisioning in NFV-Enabled Cloud/MEC RAN Architectures,” *Applied Sciences*, vol. 8, no. 12, 2018. [Online]. Available: <https://doi.org/10.3390/app8122614>
- [114] T. Taleb, A. Ksentini, and P. Frangoudis, “Follow-Me Cloud: When Cloud Services Follow Mobile Users,” *IEEE Transactions on Cloud Computing*, 2016. [Online]. Available: <https://doi.org/10.1109/TCC.2016.2525987>
- [115] “ETSI NFV announces new features to its architecture to support 5G,” [Last accessed: October 2019]. [Online]. Available: <https://www.etsi.org/newsroom/press-releases/1622-2019-07-etsi-nfv-announces-new-features-to-its-architecture-to-support-5g>
- [116] **J.-J. Pedreno-Manresa**, J.-L. Izquierdo-Zaragoza, and P. Pavon-Marino, “Guaranteeing Traffic Survivability and Latency Awareness in Multilayer Network Design,” *Journal of Optical Communications and Networking*, vol. 9, no. 3, pp. B53–B63, 2017. [Online]. Available: <https://doi.org/10.1364/JOCN.9.000B53>
- [117] **J.-J. Pedreno-Manresa**, P. S. Khodashenas, M. S. Siddiqui, and P. Pavon-Marino, “On the Need of Joint Bandwidth and NFV Resource Orchestration: a Realistic 5G Access Network Use Case,” *IEEE Communications Letters*, vol. 22, no. 1, pp. 145–148, jan 2018. [Online]. Available: <https://doi.org/10.1109/LCOMM.2017.2760826>
- [118] J.-L. Izquierdo-Zaragoza, **J.-J. Pedreno-Manresa**, and P. Pavon-Marino, “Net2Plan: An Integrated Open-Source Framework for Multilayer Network Planning and In-Operation Simulation,” in *Proceedings of the 17th International Conference on Transparent Optical Networks (ICTON 2015)*, Budapest (Hungary), Jul. 2015. [Online]. Available: <http://dx.doi.org/10.1109/ICTON.2015.7193724>

-
- [119] J.-L. Izquierdo-Zaragoza, **J.-J. Pedreno-Manresa**, P. Pavon-Marino, O. Gonzalez de Dios, and V. López, “Dynamic operation of an IP/MPLS-over-WDM network using an open-source active stateful BGP-LS-enabled multilayer PCE,” in *Proceedings of the 18th International Conference on Transparent Optical Networks (ICTON 2016)*, Trento (Italy), Jul. 2016. [Online]. Available: <http://dx.doi.org/10.1109/ICTON.2016.7550702>
- [120] **J.-J. Pedreno-Manresa**, P. S. Khodashenas, M. S. Siddiqui, and P. Pavon-Marino, “Dynamic QoS/QoE assurance in realistic NFV-enabled 5G Access Networks,” in *Proceedings of the 19th International Conference on Transparent Optical Networks (ICTON)*. IEEE, 2017. [Online]. Available: <https://doi.org/10.1109/ICTON.2017.8025149>
- [121] **J.-J. Pedreno-Manresa**, J.-L. Izquierdo-Zaragoza, F. Cugini, and P. Pavon-Marino, “On the Benefits of Elastic Spectrum Management in Multi-hour Filterless Metro Networks,” in *Proceedings of the 22nd International Conference on Optical Network Design and Modeling (ONDM2018)*, May 2018, pp. 184–189. [Online]. Available: <https://doi.org/10.23919/ONDM.2018.8396128>
- [122] **J.-J. Pedreno-Manresa**, C. Porzi, J.-L. Izquierdo-Zaragoza, F. Cugini, and P. Pavon-Marino, “Provisioning in Metro Networks Based on Photonic Integrated WaveBand ROADMs,” in *Proceedings of 44th European Conf. Optical Communication (ECOC)*, 9 2018. [Online]. Available: <https://doi.org/10.1109/ECOC.2018.8535186>
- [123] **J.-J. Pedreno-Manresa**, P. S. Khodashenas, J.-L. Izquierdo-Zaragoza, and P. Pavon-Marino, “Improved User Experience by Dynamic Service Handover and Deployment on 5G Network Edge,” in *Proceedings of the 21st International Conference of Transparent Optical Network (ICTON)*, 7 2019. [Online]. Available: <https://doi.org/10.1109/ICTON.2019.8840228>
- [124] “Wikipedia: La enciclopedia libre,” [Last accessed: October 2019]. [Online]. Available: <https://es.wikipedia.org/>
- [125] “Internet2 Network Infrastructure Topology,” Jun. 2015, [Last accessed: October 2015]. [Online]. Available: https://www.internet2.edu/media/medialibrary/2015/08/04/NetworkMap_all.pdf
- [126] P. Pavon-Marino, S. Azodolmolky, R. Aparicio-Pardo, B. Garcia-Manrubia, Y. Poirier *et al.*, “Offline Impairment Aware RWA Algorithms for Cross-Layer Planning of Optical Networks,” *Journal of Lightwave Technology*, vol. 27, no. 12, pp. 1763–1775, Jun. 2009. [Online]. Available: <http://dx.doi.org/10.1109/JLT.2009.2018291>
- [127] “SNDlib,” [Last accessed: October 2019]. [Online]. Available: <http://sndlib.zib.de>

Appendices

Appendix A

Reference Scenarios

A.1	7-Node Spanish Network (Top7Spain)	113
A.2	9-Node Internet2 Network (Internet2)	114
A.3	14-Node National Science Foundation Network (NSFNet)	115
A.4	15-Node Atlanta Network (Atlanta)	116

In this appendix, we introduce the scenarios that were used in some of the uses cases of Parts II and III. Also, a cross-reference is provided for each network to the chapter where it was used.

Each section presents a topology snapshot, tables with traffic and link length information and some relevant data. All this information is publicly available in Net2Plan [28]. The data plane refers to the networking devices, which are responsible for efficient data switching and forwarding.

A.1 7-Node Spanish Network (Top7Spain)

Comprising the seven most populated cities in Spain (see Figure A.1), this fictitious network extracts the location and exact population of each node from [124]. Eight bidirectional links were arbitrarily deployed, and the reference traffic matrix was computed using a population-distance model ([46, Section 4.6.4]) with the following parameters: $rf = 0$, $pop_{offset} = 0$, $pop_{power} = 2$, $dist_{offset} = 0$, $dist_{power} = 1$, Euclidean distances, and maximum population/distance normalization. Results in Chapter 6 were obtained using this network.

Table A.1 shows basic information about this network, while traffic (upper triangular part) and link lengths (in km, lower triangular part) is depicted in Table A.2. All traffic and link information is symmetric.

Figure A.1: 7-node Spanish network (Spain7)

Table A.1: Basic network information (Spain7)

Number of nodes	7
Number of unidirectional links	16
Average node degree	2.29
Network diameter (using shortest path in hops)	3
Network diameter (using shortest path in km)	920.39
Network diameter (using shortest path in ms)	4.6
Total offered traffic	100

Table A.2: Traffic matrix and link lengths in km (Top7Spain)

	0	1	2	3	4	5	6
0		18.1	7.6	5.4	6.1	3.5	2.3
1	-		2	0.5	1.5	0.4	0.4
2	301.9	303.4		0.2	0.5	0.2	0.3
3	391.4	-	-		0.1	0.4	0.1
4	272.4	256.5	-	-		0.1	0.1
5	-	-	-	157.6	-		0.1
6	-	-	177.2	-	-	322.9	

A.2 9-Node Internet2 Network (Internet2)

A fraction of the Abilene research network, managed by the Internet2 consortium (see Figure A.2) [125]), as presented in [126] is depicted in this section. Symmetrical traffic (upper triangular part) and link lengths (in km, lower triangular part) information was also obtained from [126] and is shown in Table A.4. Results in Chapter 3 were obtained using this network.

A.3. 14-Node National Science Foundation Network (NSFNet)

Figure A.2: 9-node Internet2 network (Internet2)

Table A.3: Basic network information (Internet2)

Number of nodes	9
Number of unidirectional links	26
Average node degree	2.89
Network diameter (using shortest path in hops)	4
Network diameter (using shortest path in km)	4116
Network diameter (using shortest path in ms)	20.58
Total offered traffic	1000

Table A.4: Traffic matrix and link lengths in km (Internet2)

	0	1	2	3	4	5	6	7	8
0		16.1	13.8	8.7	8.6	8.6	7.4	6.2	7.1
1	1342		16.4	11	14.2	14.3	16.5	8.9	10.6
2	913	1303		11.8	11.2	10.1	7.6	7.5	8.8
3	-	-	1330		16	17.8	12.9	12.3	14.8
4	-	1705	-	818		16.9	16.2	12.5	16.4
5	-	-	-	690	-		29.6	18.6	20.1
6	-	-	-	-	-	1400		30.6	22.2
7	-	-	-	-	-	905	278		17.8
8	-	-	-	-	1385	1045	-	700	

A.3 14-Node National Science Foundation Network (NSF-Net)

This is the reference network of the National Science Foundation, also known as Nobel US. Network topology (see Figure A.3), as well as basic data, traffic, and link information in km (see Tables A.5 to A.7), were obtained from the SNDlib [127] repository. Results in Chapters 3 and 7 were obtained using this network.

Figure A.3: 14-node National Science Foundation network (NSFNet)

Table A.5: Basic network information (NSFNet)

Number of nodes	14
Number of unidirectional links	42
Average node degree	3
Network diameter (using shortest path in hops)	3
Network diameter (using shortest path in km)	4500
Network diameter (using shortest path in ms)	22.5
Total offered traffic	4000

Table A.6: Traffic matrix (NSFNet)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0		30	11.9	31.2	32.5	18.4	23.9	25.8	19	21.7	16.6	21.9	16.7	23.2
1	32.6		12.9	29.8	15.8	18.6	26.3	17.5	15.1	11.7	19.3	18.7	11.4	17.1
2	21.1	26.9		30.7	17.1	26.1	10.5	24.6	12.3	26.2	23.1	15.9	17.2	12.6
3	33.1	23.6	27.2		14.7	15.2	14.7	20.1	22.2	21.1	17	11.1	9.5	24.7
4	17.7	11.8	12.9	28.2		14.4	35.4	22.9	27.4	25.4	20.1	23.2	20.5	20.5
5	19.9	11	20.8	15.2	30.6		23.4	21.5	18.5	17	22.2	16.4	22.2	24.9
6	29.4	15.6	23.2	21.9	18.5	34.5		25.3	13	26.4	26.1	14.3	31.4	21.1
7	22.5	9.1	22.8	25.5	32.1	32.5	34.3		26.5	36.8	33.1	26.7	29.9	36.5
8	12.9	14.8	21	27.5	15	16.1	30.2	14.6		20.7	29.5	36	32	14.8
9	23.7	20	10.9	15.3	15	15.3	21.3	20.5	32		29.5	26.3	21.2	27.7
10	20.5	9.9	16.5	14.7	32.5	27.9	16.3	15	41.4	20.6		41.9	28.8	19.2
11	10.6	8.3	15.2	24.6	18.7	14	14.7	29.8	36.6	25.3	19.2		36.7	33.7
12	11.3	16.6	13.4	12.6	16	21.2	14.9	27.5	22.4	27.3	30.5	15.7		22
13	18.2	17.2	10.9	23.5	16.5	29.7	21.2	37.1	28.9	23.6	35.9	21.4	38	

A.4 15-Node Atlanta Network (Atlanta)

Based on a metropolitan area network in Atlanta (United States), and obtained from SNDlib [127], the topology is presented in Figure A.4 (the node locations are fictional). Basic data, traffic matrix and link lengths in km are shown in Tables A.8 to A.10. Results in Chapter 3 were obtained using this network.

A.4. 15-Node Atlanta Network (Atlanta)

Table A.7: Link lengths in km (NSFNet)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0		1100	1600	-	-	-	-	2800	-	-	-	-	-	-
1	-		600	1000	-	-	-	-	-	-	-	-	-	-
2	-	-		-	-	2000	-	-	-	-	-	-	-	-
3	-	-	-		600	-	-	-	-	-	2400	-	-	-
4	-	-	-	-		1100	800	-	-	-	-	-	-	-
5	-	-	-	-	-		-	-	-	1200	-	-	-	2000
6	-	-	-	-	-	-		700	-	-	-	-	-	-
7	-	-	-	-	-	-	-		700	-	-	-	-	-
8	-	-	-	-	-	-	-	-		900	-	500	500	-
9	-	-	-	-	-	-	-	-	-		-	-	-	-
10	-	-	-	-	-	-	-	-	-	-		800	800	-
11	-	-	-	-	-	-	-	-	-	-	-		-	300
12	-	-	-	-	-	-	-	-	-	-	-	-		300
13	-	-	-	-	-	-	-	-	-	-	-	-	-

Figure A.4: 15-node Atlanta network (Atlanta)

Table A.8: Basic network information (Atlanta)

Number of nodes	15
Number of unidirectional links	44
Average node degree	2.93
Network diameter (using shortest path in hops)	5
Network diameter (using shortest path in km)	6000
Network diameter (using shortest path in ms)	30
Total offered traffic	136.73

Table A.9: Traffic matrix (Atlanta)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0		6	0.8	0.6	0.4	1.6	0.4	0.6	0.5	0.6	0.3	0.7	0.3	1.6	0.7
1	7.1		2.6	2.2	0.8	6.1	1.8	4.9	0.3	2	0.8	1.1	0.6	0.4	1.9
2	1	3.2		0.3	0.2	0.3	0	1.1	0.1	0.1	0.1	0.1	0.1	0	0.4
3	0.7	2.7	0.4		1	0.9	0.1	0.2	0.1	0.2	0.1	0.1	1.5	0.2	0.2
4	0.5	0.9	0.3	1.2		0.1	0	0.1	0.1	0.1	0	0.1	0.1	0	0.1
5	1.9	7.3	0.3	1.1	0.1		0.1	0.9	0.1	0.6	0.2	0.1	0.2	0.2	0.3
6	0.5	2.2	0	0.1	0	0.1		0.1	0.3	0.2	0.1	0.1	0.1	0.1	0.2
7	0.7	5.9	1.4	0.2	0.1	1	0.1		0.2	0.2	0.1	0.1	0.2	0.2	0.5
8	0.6	0.3	0.1	0.1	0.1	0.1	0.3	0.3		0.2	0.1	0.3	0.1	0.1	0.3
9	0.7	2.3	0.1	0.3	0.1	0.7	0.2	0.2	0.3		0.4	1.2	0.9	1.1	0.7
10	0.3	0.9	0.1	0.2	0	0.2	0.1	0.1	0.1	0.5		0.2	0.6	0.4	0.2
11	0.8	1.3	0.2	0.2	0.1	0.2	0.1	0.2	0.3	1.4	0.3		0.1	0.1	0.9
12	0.4	0.7	0.1	1.8	0.1	0.2	0.1	0.2	0.1	1.1	0.7	0.1		0.3	0.2
13	1.9	0.5	0	0.2	0	0.3	0.1	0.3	0.1	1.4	0.5	0.1	0.4		0.1
14	0.8	2.2	0.5	0.2	0.1	0.3	0.2	0.6	0.3	0.8	0.2	1.1	0.2	0.1	

Table A.10: Link lengths in km (Atlanta)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0		-	-	-	-	1078.6	1188.4	1085.3	-	-	-	-	-	-	-
1	-		685.4	-	1638.9	1181.1	-	-	-	-	-	-	-	-	-
2	-	-		-	1993.4	-	-	1429.9	-	-	-	-	-	-	-
3	-	-	-		1305.1	1050	-	-	-	-	-	-	-	-	-
4	-	-	-	-		-	-	-	-	-	-	-	-	-	-
5	-	-	-	-	-		-	-	-	-	-	-	1479.1	-	-
6	-	-	-	-	-	-		-	-	852.9	-	-	-	1319.3	-
7	-	-	-	-	-	-	-		827.3	-	-	-	-	-	1489.7
8	-	-	-	-	-	-	-	-		1715.6	-	1255.3	-	-	1585.5
9	-	-	-	-	-	-	-	-	-		-	1187.4	-	-	-
10	-	-	-	-	-	-	-	-	-	-		-	1693.5	917.5	-
11	-	-	-	-	-	-	-	-	-	-	-		-	-	-
12	-	-	-	-	-	-	-	-	-	-	-	-		1912.5	-
13	-	-	-	-	-	-	-	-	-	-	-	-	-		-
14	-	-	-	-	-	-	-	-	-	-	-	-	-	-	

This work has been funded by the following entities:

- Universidad Politécnica de Cartagena through different mobility actions, namely “Programa de Formación del PDI a través de la Movilidad. Convocatoria 2017” and “Programa de Formación del PDI a través de la Movilidad. Convocatoria 2018”.
- Spanish Ministry of Education, Culture and Sports through a fellowship from the “Programa de Ayudas para Formación de Profesorado Universitario. Convocatoria 2014” (ref. no. FPU2014/04227)
- Spanish Ministry of Economy and Competitiveness (MINECO), the State Research Agency (AEI) and the European Regional Development Fund through the projects ONOFRE, ONOFRE-2 and ELASTIC (ref. no. TEC2014-53071-C3-1-P, TEC2017-84423-C3-1-P and TEC2015-71932-REDT respectively).
- European Commission: Horizon 2020 European Union funding for Research & Innovation through the project Metro-Haul (ref. no. 761727).
- Several companies through technology transfer contracts regulated by the Article 83 of Spanish Universities Act 6/2001

Universidad
Politécnica
de Cartagena

European
Commission

Horizon 2020
European Union funding
for Research & Innovation