Politecnico di Milano
Dipartimento di Elettronica, Informazione e Bioingegneria

DOTTORATO DI RICERCA IN INGEGNERIA DELL'INFORMAZIONE

Advance Optical Routing Techniques in the Software Defined Era

Doctoral Dissertation of:
Rodolfo Alvizu

Advisor:
Prof. Guido Maier

Tutor:
Prof. Paolo Giacomazzi

Supervisor of the Doctoral Program:
Prof. Andrea Bonarini

2016 – Cycle XXIX

To my family

Abstract

Paradoxically, with an ever-increasing traffic demand, today transport-network operators experience a progressive erosion of their margins. The operational complexity, the use of manual configuration, the static nature of current technologies together with fast-changing traffic profiles lead to: inefficient network utilization, over-provisioning of resources and very high Capital expenditures (CapEx) and Operational expenses (OpEx).

The alarms of change are set for network operators, and Software Define Networking (SDN) is accepted as a concrete solution to reduce CapEx and OpEx and to boost network innovation. The implementation of SDN in transport networks (T-SDN) gained big momentum in the last years, however in the networking industry, the transport network will be perhaps the last segment to embrace SDN, mainly due to the heterogeneous nature and complexity of the optical equipment composing it.

This thesis starts with a deep dive into a fascinating technological adventure that provides an organic analysis of the T-SDN development and evolution. Our work is the first to consider contributions from the whole transport network ecosystem composed by: academic research, standardization bodies, industrial development, open source projects and alliances among them. After creating a comprehensive picture of T-SDN, we provide an analysis of many open issues that are expected to need significant future work, and give our vision in this path towards a fully programmable and dynamic transport network.

Then, assuming the deployment of T-SDN technologies, we propose operational research formulations and heuristic algorithms for two advanced routing techniques: dynamic optical routing and multipath optical routing. Among our contributions we can highlight the novel use of spatio-temporal (tidal) traffic demand prediction to improve the dynamic routing decisions, and the first-time proposal of two techniques to mitigate the *differential delay*, recognized as the main problem of multipath routing: 1) Differential delay equalization using unconventional routing cycles (e.g., loops), which exploits the nature of optical communications in which delays are deterministic, 2) Transparent differential delay compensation technique based on fiber delay lines, that avoids the use of conventional electronic buffering.

Finally, based on our background regarding T-SDN and advanced routing algorithms, we implemented two SDN use cases. The first use case is a T-SDN network Orchestrator for multi-domain networks, with the following capabilities: cross-domain segment routing, application-aware path selection, and multipath routing. The second use case is a path manager and the related automated testbed for evaluating Multipath TCP (MPTCP), that allowed us to deploy differential delay-aware algorithms and to experimentally demonstrate the impact of delay and differential delay on the overall performance of MPTCP.

List of Publications

P1 **R. Alvizu** and G. Maier. Can open flow make transport networks smarter and dynamic? An overview on transport SDN. in *proceedings of IEEE SaCoNeT*, Jun. 2014. p. 1-6.

The material of this publication contributes to Chapter 2.

P1 **R. Alvizu**, G. Maier, N. Kukreja, A. Pattavina, R. Morro, A. Capello and C. Cavazzoni, The big challenge: Software Defined Networking for Transport Data Networks. submitted for publication to *IEEE Communication Surveys Tutorials*.

The material of this publication contributes to Chapter 2.

P3 **R. Alvizu**, X. Zhao, G. Maier, Y. Xu and A. Pattavina, Energy aware optimization of mobile metro-core network under predictable aggregated traffic patterns, in *proceedings of IEEE ICC*, Kuala Lumpur, pp. 1-7, May 2016.

The material of this publication contributes to Chapter 3.

P4 **R. Alvizu**, X. Zhao, G. Maier, Y. Xu and A. Pattavina, Energy efficient dynamic optical routing for mobile metro-core networks under tidal traffic patterns, *IEEE/OSA Journal of Lightwave Technology*, vol.PP, no.99, pp.1-1.

The material of this publication contributes to Chapter 3.

P5 **R. Alvizu**, G. Maier, M. Tornatore and M. Pióro, Differential delay constrained multipath routing for SDN and optical networks, *Elsevier Electronic Notes in Discrete Mathematics*, vol. 52, pp. 277-284, Jun. 2016.

The material of this publication contributes to Chapter 4.

P6 **R. Alvizu**, J. Valencia and G. Maier, Multipath Optical Routing with Compact Fiber Delay Line-based Differential Delay Compensation, in *proceedings of European Conference on Networks and Optical Communications (NOC)*, Lisbon, Jun. 2016.

The material of this publication contributes to Chapter 5.

P7 N. Kukreja, **R. Alvizu**, A. Kos, G. Maier, R. Morro, A. Capello and C. Cavazzoni, Demonstration of SDN-Based Orchestration for Multi-Domain Segment Routing Networks, in *proceedings of ICTON*, Trento, Jul. 2016.

The material of this publication contributes to Chapter 6.

P8 N. Kukreja, G. Maier, **R. Alvizu** and A. Pattavina, SDN based automated testbed for evaluating multipath TCP, *in proceedings of IEEE ICC*, Kuala Lumpur, pp. 718-723, May 2016.

The material of this publication contributes to Chapter 7.

Contents

List of Figures **x**

List of Tables **xiv**

1 Introduction **1**
1.1 SDN and the need for T-SDN 4
1.2 Dynamic optical routing and tidal traffic predictability 5
1.3 Multipath routing 6
1.4 Contribution and Outline of the Thesis 7

2 Transport SDN as an enabler for advance routing techniques **11**
2.1 Introduction 11
2.2 Related works 13
2.2.1 Contribution 14
2.3 Transport SDN (T-SDN) 14
2.3.1 Formal definition of Transport SDN (T-SDN) 14
2.3.2 Enabling transport network technologies 15
2.3.3 Challenges of T-SDN 16
2.3.4 T-SDN classification 19
2.4 Research Efforts 20
2.4.1 Monolithic architecture: SDON 20
2.4.2 Hierarchical architecture: HT-SDN 24
2.4.3 Virtualization 28
2.5 Other Research Developments 33
2.5.1 Protection and Restoration 33
2.5.2 Segment Routing 35
2.5.3 Emulation 35
2.6 T-SDN Standardization Efforts 36
2.6.1 Open Networking Foundation (ONF) 36
2.6.2 Optical Internetworking Forum (OIF) 38
2.6.3 IETF 39
2.6.4 ITU-T 42
2.7 Transport APIs Standardization 42
2.7.1 IETF Transport APIs 43

2.7.2 OIF Transport API . . . 44
2.7.3 ONF Transport APIs (TAPI) & Common Information Model . . . 45
2.7.4 OPEN ROADM APIs . . . 46
2.8 Main Open Source SDN Control frameworks . . . 47
2.8.1 OpenDaylight (ODL) . . . 47
2.8.2 Open Network Operating System (ONOS) . . . 48
2.9 Vendor Solutions . . . 51
2.10 T-SDN Open Issues . . . 52
2.10.1 Control plane architecture . . . 53
2.10.2 Abstractions . . . 53
2.10.3 Common Information model . . . 54
2.10.4 North bound Interface (NBI) APIs . . . 54
2.10.5 South Bound Interface (SBI) . . . 55
2.10.6 Scalability and reliability: the distributed controller . . . 55
2.10.7 Orchestration . . . 56
2.10.8 Algorithms . . . 57
2.10.9 T-SDN, NFV and security . . . 57
2.10.10 Migration Path towards T-SDN . . . 58
2.11 Concluding Remarks . . . 59
2.11.1 Summary . . . 59
2.11.2 Final Comments . . . 60

3 Energy efficient dynamic optical routing for mobile metro-core networks under tidal traffic 61
3.1 Introduction . . . 62
3.2 Related works . . . 63
3.2.1 Contribution . . . 64
3.3 Human mobility patterns and tidal traffic demand generation in MCN . . . 64
3.3.1 The mobile carrier network (MCN) . . . 64
3.3.2 Human Mobility Patterns . . . 66
3.3.3 Per Cell traffic generation . . . 67
3.3.4 Aggregated Tidal Traffic Generation . . . 69
3.4 Offline Mobility and Energy aware Optimization Procedure . . . 70
3.4.1 Virtual Wavelength Path (VWP) . . . 72
3.4.2 Wavelength Path (WP) . . . 73
3.4.3 Complexity Analysis . . . 74
3.5 Online Dynamic Bandwidth allocation . . . 75
3.5.1 Optimal weights for VWP model . . . 75
3.5.2 Optimal weights for WP model . . . 77
3.5.3 Link-disjoint path-pairs computation . . . 79
3.5.4 Reconfiguration Time Points Scheduling . . . 80
3.5.5 Complexity Analysis . . . 80
3.6 Results . . . 81

3.6.1 Power Consumption Models . . . 81
3.6.2 Offline Numerical results . . . 81
3.6.3 Online results . . . 83
3.7 Concluding Remarks . . . 85

4 Differential delay constrained multipath routing for SDN and optical networks 87
4.1 Introduction . . . 88
4.2 Related works . . . 89
4.2.1 Contribution . . . 89
4.3 Optimization models . . . 90
4.3.1 3D*-Loop-Free formulation (3D*-LF) . . . 91
4.3.2 3D*-Loops-in-path formulation (3D*-LIP) . . . 92
4.4 Results . . . 93
4.5 Concluding Remarks . . . 95

5 Multipath Optical Routing with Compact Fiber Delay Line-based Differential Delay Compensation 97
5.1 Introduction . . . 98
5.2 Related works . . . 99
5.2.1 Contribution . . . 100
5.3 Network Scenario . . . 101
5.4 Optimization models . . . 103
5.5 Results . . . 105
5.6 Concluding Remarks . . . 108

6 Demonstration of SDN-Based Orchestration for Multi-Domain Segment Routing Networks 111
6.1 Introduction . . . 111
6.2 Related works . . . 112
6.2.1 Contribution . . . 113
6.3 Hierarchical Architecture . . . 113
6.4 Implementation and use cases . . . 114
6.4.1 Equal Cost Multipath (ECMP) use case . . . 115
6.4.2 Application-aware SR use case . . . 115
6.4.3 Multi-domain use case . . . 116
6.5 Results . . . 117
6.6 Concluding Remarks . . . 118

7 SDN based Automated Testbed for Evaluating Multipath TCP 121
7.1 Introduction . . . 121
7.2 Related works . . . 124
7.3 SDN hierarchical architecture . . . 125
7.4 Test-bed implementation . . . 127

7.5 Results 128
7.6 Concluding Remarks 132

8 Concluding Remarks **135**

Bibliography **139**

List of Figures

1.1 Legacy network architecture, composed by equipment that integrates proprietary specialized forwarding hardware, proprietary operating system and predefined set of distributed control and management features. 4
1.2 Basic SDN network architecture, composed by commodity-hardware packet forwarding devices, a logically centralized controller that collects network state and push forwarding rules. Control and management features are implemented as applications. 5

2.1 Legacy transport network architecture. 18
2.2 First proposal of SDN and GMPLS integration via ASON-UNI (SDN/UNI-GMPLS) [1]. The UNI offers a big switch abstraction with end points control and visibility. 22
2.3 Architecture of the first pure OpenFlow agent-based model that provides full visibility and control of optical domains [2]. The agent (virtual OpenFlow switch; VOFS) interacts with the optical device management interface using TL1 and abstracts the optical node. . 23
2.4 The NETCONF-based controller [3]. Agents on top of optical NEs are composed by YANG [4] database, that provides proper abstraction of optical NEs. YANG databases are populates with the link layer discovery protocol (LLDP) over an optical supervisory channel (OSC). 24
2.5 Hierarchical T-SDN (HT-SDN) architecture. 25
2.6 All-OpenFlow HT-SDN architecture [5]. 25
2.7 Hierarchical integration of SDN and GMPLS control planes over packet and optical domains using OpenDaylight and AS-PCE [6]. . 26
2.8 HT-SDN architecture based stateful hierarchical PCE (SH-PCE) [7]. 27
2.9 ABNO-based orchestration over heterogeneous OPS and OCS domains [8]. 27
2.10 South bound virtualization architecture for the multi-domain scenario. 29
2.11 Central virtualization architecture for the multi-domain scenario. . 30
2.12 North bound virtualization architecture for the multi-domain scenario. 31
2.13 Reference Hierarchical Control Architecture proposed by ONF [9]. 38

2.14 HT-SDN architecture proposed by OIF [10]. 39
2.15 Generic ABNO architecture from RFC 7491 [11]. 41

3.1 Reference mobile carrier network (MCN) architecture. 66
3.2 Synthetic topology of the Chinese city based on real location data and a conservative dimensioning on the peak. 67
3.3 Average number of users $N_{c,t}$ within 4 cells of the Chinese city. . . 68
3.4 Generated traffic load $L_{c,t}$ samples within 4 cells of the Chinese city. 69
3.5 Predictable aggregated tidal traffic demands (down-link direction) $H_{j,t}^{DL}$. 71
3.6 Equivalent layered-graph for the physical topology shown in the top left corner, and considering only two dummy nodes $1^s \in \mathcal{V}^s$ and $5^d \in \mathcal{V}^d$ where a demand from OXC1 to OXC5 can be mapped. . . 78
3.7 Optical layer power consumption of the mobile metro-core network under predictable aggregated traffic using three of-line resource allocation methods: Static-VWP, Hourly-VWP and Scheduled-VWP (Time point optimized). 82
3.8 Optical layer power consumption of the mobile metro-core network under predictable aggregated traffic using three of-line resource allocation methods: Static-WP, Hourly-WP and Scheduled-WP (Time point optimized). 83
3.9 Energy dissipation in the mobile metro-core network for three different methods: VWP off-line Optimization (under predictable aggregated traffic), VWP on-line matheuristic and VWP on-line heuristic. 84
3.10 Energy dissipation in the mobile metro-core network for three different methods: WP off-line Optimization (under predictable aggregated traffic), WP on-line matheuristic and WP on-line heuristic. . 84

4.1 Comparison of 3D-LIP and 3D-LF in a 10-node network 92
4.2 14-node test network 94
4.3 Comparison of 3D-LIP and 3D-LF in a 14-node test network based on Maximum K. 94
4.4 Comparison of 3D-LIP and 3D-LF in a 14-node test network, based on Minimum L. 95

5.1 Common approach: centralized-DDC. 99
5.2 Distributed electronic-DDC. 100
5.3 Proposed distributed transparent-DDC (with FDLs). 101
5.4 Optical node composed by: Optical cross connector (OXC), FDL module for Transparent-DDC, Electronic-DDC collocated with optical regeneration, and core router. 102
5.5 Transparent & electronic-DDC vs FDL Module length G, using FDL-RDBuff in NSFNET. 106

5.6 Transparent & electronic-DDC vs FDL Module length G, using FDL-DBuff in Polish Network. 107
5.7 Total Buffer size of Electronic-DDC when solving CBuff, DBuff, FDL-CBuff, FDL-DBuff and FDL-RDBuff for Polish and NSFNET networks. 108
5.8 Total FDL size of transparent-DDC when solving FDL-CBuff, FDL-DBuff and FDL-RDBuff for Polish and NSFNET networks. 108

6.1 General multi-domain hierarchical SDN Architecture. 114
6.2 Forwarding Information Base (FIB) generated by router 1 in Domain 1 of the general architecture. 116
6.3 Multi-domain TE use case. 118
6.4 Part of an Explicit Route Object (ERO) example. 118

7.1 General SDN Architecture with MPTCP python application for path management, and OpenDaylight controller with the added MPTCP OSGI module that sends notification to the MPTCP application. . 126
7.2 Differential delay impact on throughput decomposition for short flows connections (MPTCP path engine). 129
7.3 Differential delay impact on throughput decomposition for Elephanft flows (MPTCP path engine). 130
7.4 Effect of differential delay and delay for RR scheduler. 130
7.5 Effect of differential delay and delay for default scheduler (lowest round trip time first). 131
7.6 Combined effect of differential delay and delay for RR scheduler. . 131
7.7 Combined effect of differential delay and delay for default scheduler (lowest round trip time first). 132

List of Tables

2.1 Transport SDN Characteristics 17
2.2 Time-line of research activities for enabling T-SDN with SDON . . 21
2.3 Time-line of research activities for enabling T-SDN with HT-SDN 21
2.4 Timeline of research activities on T-SDN Virtualization 22
2.5 Timeline of Standardization on T-SDN architecture, control plane and south bound interfaces 37
2.6 Timeline of Transport API standardization efforts 37
2.7 Standardization of Northbound transport APIs (NB-TAPIs) 43
2.8 Vendor Transport SDN solutions - optical *blackbox* 52
2.9 Transport SDN *white-box* solutions 53

3.1 Optimal Link Weights ($c_{e,\hat{r},d,t}$) for VWP 77
3.2 Optimal Virtual Link Weights ($c_{e',\hat{r},d}$) for WP 79
3.3 Energy dissipation and disruption rate trade-off 82
3.4 On-line results - Average Energy consumption increase 85

5.1 Evaluated DDC techniques 103

6.1 Traffic paths of example in Fig. 6.3. 117

Introduction 1

The technological evolution that the world is experiencing is maintained by acquisition, storage, processing and exchange of information. Bit streams of information over the Internet Protocol (IP), known as IP traffic continuously crosses the world to reach: end users through high definition (HD) screens (either in mobile and fixed devices), connected machines (Internet of Things), and Information Technologies (IT) infrastructures, *i.e.*, public or private datacenters and the Cloud infrastructures. IP traffic is powered by ubiquitous Internet connectivity provided by Communication Service Providers (CSP)s. By the end of 2016, the annual global IP traffic will finally reach the zettabyte ($1.1 \cdot 10^{12}$ gigabytes) threshold, and with an expected compound annual growth rate of 22%, it will reach 2.3 zettabytes by 2020 [12].

CSPs face an ever increasing demand for resources that does not correspond to the increase of their revenues. To preserve their margins, they must be able to substantially reduce capital expenditures (CapEx) and operational expenses (OpEx). However, current technologies and architectures are constrained by their operational complexity and static nature.

The common practice in CSP networks is to perform a static resource allocation which meets the peak-hour demand. This method leads to poor network and energy efficiency, as outside the peak hour resources will be over-provisioned. As a consequence, the Internet has become one of the major energy consumer in the world [13].

Energy efficiency has become a must when designing and proposing new network technologies and protocols, because of the direct impact in the OpEx and the environmental problems due to the green house effect.

There is a trade-off between network performance, that must meet the stricter quality of service (QoS) requirements of emerging Internet services, and the reduction of CapEx, OpEx and energy consumption. Thus the networking

techniques must be carefully designed in order to meet a fare trade-off between cost, energy consumption and performance.

CSPs have set the alarms of change and have started to look at new technological solutions. In this work we focus on Software Defined Networking (SDN), that is becoming an established trend for management and control of telecommunication network infrastructure [14]. By separation of control and data plane, SDN brings networking back to centralized control, allowing applications to program the network forwarding rules through logically centralized controllers. SDN brings a promising solution to introduce network programmability and accelerate networking innovation. For instance, network programmability allows to implement intelligent dynamic optimization methods to improve performance and energy efficiency, and to create new revenue streams through fast deployment of innovative services.

As we will show in this thesis, there are strong reasons to believe that SDN can actually be a cost-slashing technology for CSPs. In fact, it has already proven to be such for datacenter operators and at the edge of transport networks (e.g., campus networks), thanks to the relative uniformity of switches used. However, the implementation of SDN in transport network, known as T-SDN (Transport Software Defined Networking), is more difficult because of the heterogeneous nature of the equipment composing the transport networks. Transport networks are composed by multiple domains delimited by technologies and vendor islands, cover multiple layers from IP to optical, and must deal with optical constraints. In consequence, T-SDN still represents a major challenge.

The present study starts by investigating T-SDN under the point of view of large transport-network operators such as TIM (the Italian incumbent operator). The comprehensive analysis of T-SDN that we provide is actually based on the experience of collaboration in a joint research project with TIM. We describe the complete picture and historical evolution of T-SDN developments taking into account the whole ecosystem of transport network players composed by: academic research, standardization bodies, large international projects, industrial vendors, and open source and industrial alliances.

Then we assume a scenario in which T-SDN is already implemented. This assumption allows us exploring the benefits of two advanced routing techniques: dynamic optical routing and multipath routing. These techniques exploit the centralized programmability and view of the network to overcome implementation and performance problems of advanced routing.

The CSPs plan the network capacity according to peak rates. However the peak to average ratio is growing, exacerbating the need for dynamic resource allocation techniques. In 2015, peak hour Internet traffic grew 51 %, while the average traffic grew at 29 % [12]. It is expected that between 2015 and 2020, global peak-hour Internet traffic will grow at a compound annual growth rate (CAGR) of 36 %, compared with 25 % for average Internet traffic.

SDN allows the dynamic movement of capacity to handle quickly-changing traffic demands, avoiding resource over-provisioning and sub-utilization without degrading the quality of provided services. Within the (FP7 Marie Curie IRSES)

Mobile Cloud project, we proposed the use of traffic pattern predictability to improve dynamic optical routing techniques. More specifically, we proposed off-line optimization and on-line matheuristic methods, that use tidal traffic pattern predictability to improve load-adaptive dynamic optical routing. Tidal traffic refers to the repetitive patterns with spatio-temporal variations that recalls the rise and fall of the sea levels, knows as tides.

In this work we have also analysed the service disruption due to changes in the routing, that can affect the performance of services using the network. We propose a heuristic method to achieve a fair trade-off between energy efficiency and service disruption of our load adaptive routing techniques.

Then we focus on Multipath (MP) routing, an effective technique for applications imposing stringent requirements on bandwidth, delay and availability. Even though MP routing is a promising technique, it never took off in networking due to the delay difference between paths of the MP connection, *i.e.* the *differential delay* (DD). In presence of DD, the destination receives a disordered version of the original packet sequence. Thus DD affects the performance of the MP connections by increasing *jitter* and packet losses due to packet reordering tasks and buffer overflows. Assuming an SDN network, we have proposed novel optimization methods for two MP routing problems:

- MP routing with disjoint differential delay constraint problem (the 3D problem).
- MP routing with differential delay compensation (DDC) problem.

From the 3D problem experience we started to look at the DDC problem, and the study of DDC problem for transparent optical networks led us to propose, for the first time, the use of compact fiber delay lines (FDL)s to perform distributed all-optical DDC.

The comprehensive analysis of T-SDN, and the proposal of advanced routing techniques allowed us to acquire the necessary background expertise to start the implementation of T-SDN solutions. In the final part of this thesis we present the development of a hierarchical control plane architecture for multi-domain segment routing networks based on an SDN Orchestration platform.

Then we also describe the implementation of an SDN-based path manager and the related automated testbed for evaluating Multipath TCP (MPTCP). The path manager-application reactively create optimum paths for MPTCP taking into account: path-disjointness, delay, DD and available bandwidth. Using the automated testbed we present an in-depth analysis of different scheduling algorithms to understand how delay and DD affect the overall performance of MPTCP protocol.

In the following subsections we introduce the technological solutions covered in this study.

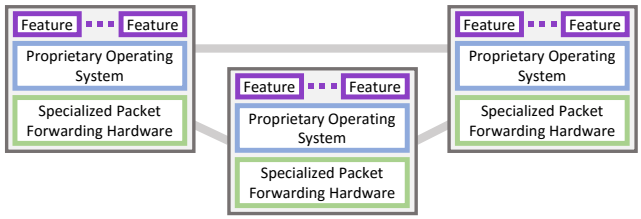

Figure 1.1: Legacy network architecture, composed by equipment that integrates proprietary specialized forwarding hardware, proprietary operating system and predefined set of distributed control and management features.

1.1 SDN and the need for T-SDN

Fig. 1.1 depicts the legacy network architectures, which are based on purpose- and vendor-specific systems composed by highly-integrated and specialized forwarding-chips, proprietary operating systems and predefined features. In order to apply new network policies, an operator has to configure each device using vendor-specific command line interfaces (CLI)s. To provide a new feature, an operator may wait for a long period before the device vendors release a software upgrade that supports it. The distributed network intelligence makes hard to understand the current state of the network and to apply new forwarding rules. The integrated system presented in Fig. 1.1 is "ossified", *i.e.* imposes a challenge towards innovation and network evolution. A clear example is the migration from IPv4 to IPv6 that after more than 10 years is not near to be fully accomplished.

On the other hand, SDN is based on open systems, where purpose and features of the system are provided through development of software and applications. The Open Networking Foundation (ONF) [15], an organization dedicated to the promotion and adoption of SDN, defines it as a programmable network architecture where the control plane is separated from the data plane (forwarding hardware) as depicted in Fig. 1.2.

By decoupling control and data planes, the network intelligence and state can be logically centralized, the forwarding infrastructure can be conveniently abstracted to the applications plane, and innovation is boosted independently at each plane [16].

The evolution of SDN is motivated by three main markets: enterprises, cloud service providers (datacenters) and telecommunication service providers [17]. Among them, datacenters and enterprises (which mostly use networks based on packet switching) have experienced a fast development of SDN solutions and worldwide SDN deployments [18, 14]. Data centers and big companies like Google were the first to deploy SDN-based solutions [19].

The telecommunication service providers are far behind SDN deployments due to the challenges and complexity of transport networks.

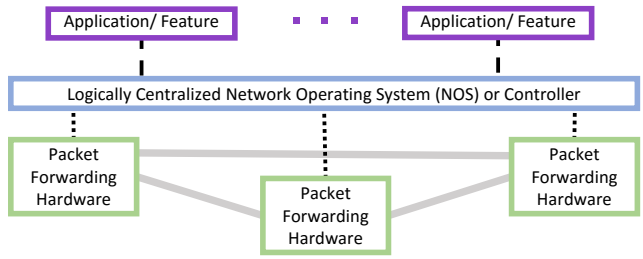

FIGURE 1.2: Basic SDN network architecture, composed by commodity-hardware packet forwarding devices, a logically centralized controller that collects network state and push forwarding rules. Control and management features are implemented as applications.

Transport networks are composed by heterogeneous multi-layer, multi-domain and multi-vendor architectures. While SDN was conceived for packet-oriented layers, the transport network also involves circuit-oriented layers and must control the complexity of optical and/or wireless domains. Moreover, the optical network elements have vendor-specific implementations that lead to an heterogeneous data plane that is not easily represented with OpenFlow semantics.

Transport SDN (T-SDN) is an SDN-based architecture for the control and management of transport networks, that could involve multi-layer, multi-domain and multi-vendor scenarios.

Telecommunication operators and transport service providers have strong interests in deploying T-SDN to migrate their transport infrastructure from an “ossified" and static architecture to a dynamic and programmable system. However, T-SDN still represents a major challenge and there is no consolidated commercial solution nor stable standards so far. Some optical vendors, in collaboration with Open Source projects, are at the initial stage of their T-SDN solutions. Standardization bodies are working to guarantee the interoperability of vendor solutions, but the standardization process is far from being completed.

1.2 Dynamic optical routing and tidal traffic predictability

In order to improve network efficiency, instead of planning the network resources for the busy-hour demand and using a static allocation, the resources can be allocated dynamically to follow the traffic load fluctuations in the network. SDN allows telecommunication service providers to gain control of their networks through application programming interfaces that facilitates the implementation of such dynamic techniques. The energy efficiency limits of networks that can adapt to traffic load variations in time was demonstrated in [20].

In this work we focus in the mobile metro-core network. The volume of mobile data traffic is by far smaller in comparison with its fixed counterpart, but it is growing at a faster pace: global mobile data traffic was 5 % of total IP traffic in 2015, and will be 16 % of total IP traffic by 2020 [12]. The popularity of smart-phones, the advent of HD-resolution mobile-terminal screens, mobile cloud services and the Internet of things are major contributions to the expected mobile data traffic 10-fold growth from 2016 to 2021 [21]. For instance, in 2016 commercial LTE networks are starting to support downlink peak data speeds of 1 Gbps [21], and among the identified 5G requirements a 1000-fold growth of bandwidth per unit area [22].

Most of the works on resource allocation and energy efficiency optimization take into account only the temporal fluctuation of an overall traffic demand. However, such homogeneous traffic matrix is far from the behaviour of traffic load in a metropolitan area. In this work we consider traffic from several different locations in the urban area, thus we are able to capture the spatial and temporal fluctuations of the traffic. Such three dimensional surface provides valuable information for optimization purposes.

Mobile data traffic is very dynamic; nevertheless due to the highly predictable daily movements of large populations of citizens in urban areas [23], the mobile traffic exhibits repetitive patterns with spatio-temporal variations. This behaviour has been recently compared to the rise and fall of the sea levels, known as tides. Thus it was called the tidal-traffic scenario [24].

Tidal traffic may create hot spots in the network that move in the spatio-temporal space, following a regular pattern given by the human commutation from residential areas to working areas (academic, business, industrial, medical, governmental among others). Special events may perturbate the tidal traffic, as for instance maintenance, disasters, and social events.

Therefore, the high predictability in human mobility patterns is a valuable information to optimize resource allocation and increase energy efficiency by effectively adapting to expected tidal traffic load variations, as we demonstrate in chapter 3.

1.3 Multipath routing

Historically, Internet traffic has been routed over a single shortest path: which was convenient for best-effort data traffic; but it is not always suitable for today's scenario. For instance, multimedia applications based on ultra-high definition digital media formats (i.e., 4K and 8K resolution), inter datacenter communications and e-science applications can require bandwidth higher than what is available in a single link, even when the link is provided by an optical wavelength channel.

Multipath (MP) routing allows to split a single connection over multiple sub-channels that can be forwarded over physically separated paths. MP

routing broadens the possibilities to set up high bandwidth connections and to improve the network resource utilization.

Current networks already offer MP capabilities, e.g., multi-homed datacenters and enterprise networks, and end-devices with multiple interfaces (Wi-Fi and cellular). MP routing is an effective networking functionality that allows to increase throughput and network resource utilization and improve resilience to possible link failures [25].

However, the common networking practice seldom exploits MP routing. The main problem of MP routing is that each path in an MP group experiences a specific delay, leading to a differential delay (DD) between the paths.

The common way of compensating the effects of DD in MP routing is to use reconstruction buffers at destination. However, this technique may require very large reconstruction buffers and may lead to poor performance in terms of jitter and packet losses. In circuit-switched optical networks there are two main approaches to mitigate the effects of DD in MP routing [1]:

- DD-constraint MP routing: that finds delay-similar paths by minimizing or constraining the DD of the MP set [27, 28, 29],
- DD-compensated MP routing (DDC): that delays the shorter paths of the MP set at intermediate nodes to meet a DD constraint, as proposed for the first time by Alicherry *et al.* [30].

The first approach is constrained by the network topology, while the second allows more flexibility to mitigate the DD, but may incur in extra costs as we will further analyse in chapters 4 and 5, respectively.

1.4 Contribution and Outline of the Thesis

This PhD research work is three-folded into the following main activities:

1. Comprehensive survey of the efforts done by academia, standardization bodies, vendors and open source projects to enable SDN in transport networks (T-SDN) with a deep analysis and classification of this new technology.
2. Proposal of novel optical routing techniques that exploit SDN programmability.
3. Implementation of SDN-based MPTCP testbed tool and multi-domain SDN Orchestrator.

By the first activity, we elaborate on the technological challenges to enable T-SDN. In the second set of activities we propose advanced routing techniques

[1] In wireless networks it is common to use scheduling schemes to mitigate the DD when using heterogeneous wireless networks (e.g., WiFi and cellular) [26, 25]. Such approach is not popular in optical networks at Gb/s because it involve expensive electronic buffering at the source node, which for Gb/s can amount to considerable sizes.

that exploit centralized network programmability and visibility enabled by T-SDN:

- Dynamic optical routing techniques based on predictable tidal traffic patterns to reduce energy consumption and avoid performance degradation of network services: Off-line optimization models and on-line matheuristics[2].
- Link disjoint multipath routing techniques that mitigates the effect of DD: 1) MP routing under DD constraint in which we analyse the trade-off between DD minimization and overall delay increase. 2) Novel energy-efficient strategies for DD compensation using fiber delay lines.

Finally, using the background expertise of the previous activities we demonstrate the implementation of two SDN solutions.

- Multi-domain[3] SDN orchestration of MPLS networks with segment routing.
- Automated test tool for multipath TCP.

More in detail, our research work is organized through the remainder of this thesis as follows:

- In Chapter 2 we start by identifying the challenges of deploying T-SDN. Then we describe and classify the research efforts on T-SDN that started with monolithic control plane architectures (SDON), and then evolved towards and hierarchical control plane architectures (HT-SDN). Another focus of research is to provide virtualization capabilities in T-SDN platforms. We then overview the standardization efforts by Open Networking Foundation (ONF), Optical Internetworking Forum (OIF), Internet Engineering Task Force (IETF) and the ITU-T. We give a special focus on the transport application programming interface (T-API), that is one of the main components to foster programmability into multi-domain transport networks. We analyse the two main open source control plane frameworks that are shaping the future of carrier-grade T-SDN controllers. We also list and compare the most influential vendors in T-SDN, including *blackbox* solutions are the more innovative and disruptive *white-box* solutions based on disaggregated and open components. Finally we analyse the many open issues that are expected to need significant future work, and give our vision in this path towards a fully programmable and dynamic transport network.

- In Chapter 3 we evaluate the energy consumption of dynamic optical routing for the mobile metro-core network use case. We use real mobile data traffic from a Chinese city to obtain spatio-temporal fluctuations on the

[2]Matheuristic is an optimization algorithm made by the interoperation of heuristics and mathematical programming.

[3]A domain refers to a SUB-network area defined by a specific: layer, vendor, data plane or control plane technology.

aggregated (tidal) traffic demand. After demonstrating the predictability of tidal traffic in mobile core networks (MCN), we proposed two Mixed-Integer Linear Programming (MILP) models that target predictable aggregated mobile tidal traffic in urban areas: a virtual wavelength path model (VWP) and a wavelength path model (WP). We then propose a suite of on-line dynamic bandwidth allocation matheuristic procedure for predictable and unpredictable tidal traffic. Our matheuristics show very small optimality gap by interacting with off-line optimization and weighted graph computation to apply minimum cost algorithms. We have also developed a method to schedule reconfiguration time points of dynamic bandwidth allocation, by capturing a trade-off between decreasing the number of reconfiguration time points (to decrease service disruption in the network) and increasing resource allocation efficiency.

- In Chapter 4 we move to another interesting routing technique based on inverse multiplexing: multipath (MP) routing. We face the problem of disjoint differential delay constraint MP routing. We then formulate two MILP models that output the optimal MP set with maximum path cardinality and minimum overall delay under a given DD constraint. In one model we exploit SDN programmability to allow the routes to use controlled in-path loops to equalize the DD. As a consequence our proposal increase the solution space and outperforms the more classical approach without loops. We introduce an iterative procedure to simplify the formulation and make the problem tractable. Our results strike the trade-off between reduced DD and increased overall end-to-end delay.

- In Chapter 5 we continue our investigation on MP routing, and given the trade-off between DD and overall delay experienced by DD-constraint MP routing, we decided to focus DD-compensated MP routing (DDC). DDC compensates the DD without penalizing the overall end-to-end delay. To the best of our knowledge we proposed, for the first time, the use of Fiber Delay Lines (FDLs) to accomplish transparent-DDC. We formulate 5 MILP models to evaluate different DDC techniques that exploit the proposed transparent-DDC. We show that the use of FDL reduce the use of expensive compensation buffers and energy consuming optical-to-electrical and electrical-to-optical (O/E-E/O) conversions. Moreover the results strike that depending on the topology dimension there is an specific combination of DDC techniques that allows to further reduce the O/E-E/O conversions.

- In Chapter 6 we move to demonstration of HT-SDN orchestration of a multi-domain network based on the knowledge gathered form Chapter 2. A hierarchical control plane architecture is created by an Orchestration application that uses the northbound interface of multiple SDN controllers over a Segment Routing (SR) network. With the Orchestrator we demonstrate cross-domain SR capabilities over SDN controlled

heterogeneous domains, application-aware path selection to guarantee QoS, and use of Equal Cost MultiPath (ECMP) capabilities. Moreover, we evaluate the scalability and flexibility of a HT-SDN architecture over multi-domain transport networks, that using northbound interface of open source controllers provides complete control of path creation and path policies to the Orchestrator.

- In Chapter 7 we report the implementation of a SDN-based multipath (MPTCP) path manager-application that can reactively create optimum paths for MPTCP. We then propose a unique automated test-bed implementation to quickly evaluate MPTCP protocol. We experimentally demonstrated the effect of delay and differential delay on the overall performance of MPTCP protocol when using Lowest RTT first and Round Robin scheduler.

  Implementation of SDN-based MPTCP path manager-application that runs complex algorithms to find suitable paths for MPTCP connections. The path manager can reactively create optimum paths for MPTCP taking into account: disjointness, delay, DD and available bandwidth. We then propose a unique automated testbed for performance evaluation, and an in-depth analysis of different scheduling algorithms to understand how delay and DD affect the overall performance of MPTCP protocol. Finally we provide numerical results and the experimental demonstration of the impact of delay and DD on the overall MPTCP performance.

- Chapter 8 draws the conclusion of the thesis and discusses some of the issues that still remain open.

Transport SDN as an enabler for advance routing techniques 2

SDN is already available at the edge of transport networks, in isolated locations like datacenters, points of presence, and even WANs are using SDN-based overlay technologies (SD-WAN). There is a growing need for SDN in the transport networks (T-SDN). However, the heterogeneous nature and complexity of optical domains represents a big challenge for T-SDN. In this Chapter we provide a deep dive into the challenges, evolutionary steps and main efforts driven by academia, industry, standardization bodies and even open source projects, to accomplish T-SDN.

2.1 Introduction

The reader of a scientific journal or the attendee of a conference in our area of networking has certainly read or heard the words “software-defined-..." hundreds of times over the last couple of years. This binomial is indeed among the most frequently spoken or written ones in the ICT technical field, probably at the same level of cloud computing, network virtualization, data center, and a few others. What is the secret of success of this magic term? As usual, the success behind a technological solution is primarily economics: we can say that softwarization of networking appeared to Internet service providers and datacenter and network operators as a glowing new hope for a future of cost reduction.

The need of cost reduction is particularly urgent for the operators of transport networks, i.e. those large networks of substantial geographical extension (regional, national, continental or even intercontinental) providing infrastructure (links and equipment) to move large and aggregated data traffic (in the

form of streams of packets). Transport-network infrastructure is expensive, especially in terms of operating costs, and must constantly be upgraded and expanded to keep the pace with the increase of traffic. This rapid growth is generated by the applications offered from the giants of Internet (the so-called over-the-tops) to users (typically, for free) and that users are more and more eager to enjoy. Therefore, taken in the vise of ever-increasing traffic and practically flat revenues from the subscribers, transport-network operators are struggling. They assist to the progressive erosion of their margins, while the same over-the-tops at the basis of their worries are capturing the largest share of the ICT market value.

So, the only possibility to preserve margins is reducing Capital expenditures (CapEx) and Operational expenses (OpEx). However, current technologies and architectures are constrained by their operational complexity and static nature, that lead to inefficient network utilization and over-provisioning of resources. As a consequence, operators have set the alarms of change and have started to look at Software Defined Networking (SDN). SDN is buzzing the networking world with the promise of increasing the programmability and innovation pace as well as reduction of OpEx and CapEx.

As we will show in this chapter, there are strong reasons to believe that SDN can actually be a cost-slashing technology, and in fact, it has already proven to be such for datacenter operators and at the edge of the transport networks. The scenario is more difficult in case of operators of transport networks, especially because of the heterogeneous nature of the equipment composing the core network, compared to the relative uniformity of switches used in the datacenters and at the edges. As we will try to convey in the following parts, the main problem is to apply the SDN concept in an environment that was not natively developed to support this new control technology.

Therefore, in this chapter we will not speak about SDN in general, but we will focus on the specific Transport-SDN (or T-SDN) scenario, investigating the topic under the use case of large transport-network operators such as TIM (the Italian incumbent operator). In fact, the present study was elaborated within the framework of a collaboration research project with TIM about preliminary investigation of Transport-SDN solutions for long-distance and optical transport networks.

As the reader will soon realize, the structure of the following is quite articulated and segmented, because the matter itself is complicated. To provide a comprehensive picture of T-SDN development we need to consider many types of system architectures and the interplay of many contributions coming from different sides: academic research, standardization bodies, large international projects, industrial development, industrial and open source alliances, and so on. We hope we will be able to guide the reader in a smooth navigation (also with an aid from the figures), providing an organic analysis.

At the end, the survey that we present in this Chapter is about a fascinating technological adventure, in which an innovation, initially underestimated, then exalted as salvific, is now currently undergoing a careful redesign process to

clear many details that before have been overlooked. All in a nutshell of years, because the time-to-market is a critical factor, under the pressure of cost reduction. And the end of the story seems still far away.

This chapter is actually based on the experience of collaboration of the author in joint research projects with TIM (Telecom Italia) about preliminary investigation of Transport-SDN solutions for long-distance and optical transport networks.

The rest of the Chapter is organized as follows: in Section 2.2 we provide an overview of the related work. In Section 2.3 we define T-SDN, and describe enabling technologies, challenges and classify T-SDN solutions by its control plane architecture. In Section 2.4 we focus on the research efforts to enable T-SDN and to provide virtualization capabilities. Then sections 2.6 and 2.7 present the activities from the main standardization bodies regarding T-SDN architectures and T-SDN north bound APIs, respectively Section 2.8 provides an insight into the two main open source frameworks (OpenDaylight [31] and ONOS [32]) for implementation of transport SDN and carrier grade controllers. Section 2.9 lists and compares the most influential vendors in T-SDN, including *blackbox* and *white-box* solutions. Section 2.10 identifies open research areas, while Section 2.11 gives our vision on the future of this topic and concludes this chapter on the path towards a fully programmable and dynamic transport network.

2.2 Related works

Multiple survey papers on SDN have been recently published [18, 33, 14, 34]. However, those works present a broad overview of SDN-technologies in multiple areas, from datacenters to wireless and optical networks. A survey and categorization of hypervisors for SDN is provided by [35].

The surveys on optical transport SDN solutions started with the first stage of its evolutionary path, where the focus was to implement SDN concepts into single domain optical networks for a unified control of optical and IP layers [36, 37, 38, 39, 40]. This necessary evolutionary step is called in literature software defined optical networks (SDON).

However, apart from being multi layer, transport networks are composed by multiple domains given by the segment (access, metropolitan and core/backbone), the technology, and even delimited by vendor islands. The Authors of [41] provide an overview that includes: monolithic control plane architectures for single domain transport networks (SDON) and the second evolutionary steps of T-SDN based on hierarchical control plane architectures for multi-domain transport networks. [41] focused on SDN research activities for optical access, metro and core networks, giving a special focus on passive optical access. The authors of [42] gives an overview of the research efforts on interworking between GMPLS and OpenFlow for control of flexi-grid networks. In [43] was presented a survey that focused on the interoperability issues of network management for

carrier-grade networks, with a focus on Multi-Technology Operations System Interface (MTOSI), NETCONF and the advent of SDN.

Another surveys focused on the specific case of SDN and OpenFlow for optical access networks [44, 45], and SDN orchestration for the multi-domain optical datacenter networking [46].

2.2.1 Contribution

This chapter provides a global view, classification and historical evolution of T-SDN developments taking into account the whole ecosystem of transport network players composed by: academic research, standardization bodies, large international projects, industrial vendors, and open source and industrial alliances.

This chapter was aimed at the acquisition of the necessary background expertise to start the implementation of T-SDN solutions. For instance, in chapter 6 we present the development of a hierarchical control plane architecture for multi-domain segment routing networks based on an SDN Orchestration platform.

This work led as by-product the elaboration of a short survey on T-SDN based on OpenFLow [38], and a comprehensive survey that provides a global view on T-SDN, submitted to IEEE communication surveys and tutorials [47].

2.3 Transport SDN (T-SDN)

Telecommunication operators and transport service providers showed strong interests in the deployment of SDN in their optical transport networks. Providers can use SDN to provide automated and efficient connectivity in order to meet new service and application need. However, the protocols and SDN-architecture extensions needed to control and manage the transport networks (called Transport SDN or T-SDN) represents a major challenge, due to the heterogeneous multi-domain (vendor, technology), multi-layer and some times even analog nature of transport networks.

2.3.1 Formal definition of Transport SDN (T-SDN)

The Optical Internetworking Forum (OIF) defines Transport SDN (T-SDN) as a subset of SDN-architecture functions comprising the transport network relevant components [10].

The ONF Optical Transport working group (ONF-OTWG), renamed as the Open Transport working group, proposed what they called OpenFlow-enabled Transport SDN-architecture as described in [9], which is mainly based on OpenFlow. At the end of 2013 the ONF published the Openflow Switch Specification v1.4.0 [48] that introduced for the first time support for optical ports. Nevertheless, the work still in progress in ONF-OTWG to define stable and standard NBI and SBI specification for SDN/OpenFlow-based T-SDN

including: extensions for management and control of optical transport [17, 9, 49, 50], and wireless transport [51]. This work focuses on the optical transport network, rather than in the wireless transport, which is an area of great interest with the advent of 5G mobile networks, the Internet of things and mobile cloud era.

The following subsection briefly describe some transport network technologies to help the reader to better understand the challenges related to the extensions of SDN principles to optical transport networks presented in subsection 2.3.3.

2.3.2 Enabling transport network technologies

The transport networks involve many different technologies across multiple layers:

- *layer 3 and 2*: IP, Ethernet, MPLS [52] and MPLS-TP [53] that provides statistical multiplexing at packet level (L2).
- *Layer 1 and 0*: at layer 1 OTN [54] that supports ODUk (Optical Channel Data Unit) [1] electrical time division multiplexing (TDM), and at layer 0 optical wavelength division multiplexing (WDM) and the new flexible grid technologies.

Traditionally, routing, signaling and protection functionalities were placed at the IP layer, and the optical layer provided static connectivity for the layer 2 and 3 devices. However, flexibility and dynamic capabilities of state-of-the-art optical devices allows to avoid the hop-by-hop IP processing by efficiently and dynamically adapting the optical connections to the traffic demands and by-passing the IP layer whenever it is possible. Optical by-pass allows to avoid the energy consumption, costs, delays and complexity of hop-by-hop IP processing. We now describe some of the optical network technologies that enable a flexible and dynamic optical layer.

- Transparent optical Networks: composed by optical devices that are capable of switching signals in the optical domain such as reconfigurable optical add-drop multiplexers (ROADM)s, wavelength cross-connects (WXC)s and photonic cross-connects (PXC)s.
- Elastic optical network (EON): consist of bandwidth variable optical cross connects (BV-OXC)s and bandwidth variable optical transponders (BVT)s. In the EONs the previously fixed WDM grid, becomes flexible (flexi-grid) by introducing spectral and modulation format flexibility, allowing lightpaths to meet the variable requirements of services and applications, as described in G.694.1 [55]. In flexi-grid the optical channels are identified by port, central frequency, frequency slot bandwidth and type of signal [56, 57].

[1] ODUk is an information structure defined in ITU-T Recommendation G.709 [54]

- Generalized Multi-Protocol Label Switching (GMPLS): GMPLS is a control plane technology (RFC 3945 [58]) proposed by the IETF to manage heterogeneous switching modes including packets, time slots, wavelengths and fibers. GMPLS is a distributed control plane based on a pool of protocols standardized by IETF (e.g. OSPF, ISIS, RSVP-TE) and it is the most used control plane in current optical transport networks. The path computation element (PCE) [59] is playing an important role in the interoperability between GMPLS and SDN.
- Network management system (NMS) and Element management system (EMS): the optical network equipment are typically controlled and managed through a centralized NMS/EMS. NMS/EMS provide a highly reliable optical resource allocation (lightpath provisioning) in a manual and semi-static fashion. The NMS computes optical reach, configures the devices, and performs monitoring to ensure proper signals quality. The NMS provides a north bound interface to the operations support system (OSS) (or applications) usually based on the simple network management protocol (SNMP), common object request broker architecture (CORBA) or extensible markup language (XML) [60].

2.3.3 Challenges of T-SDN

SDN was specifically defined for packet-switched networks at layer 2 and 3 [61]. Today, standardization for OpenFlow-based SDN is strongly supported by ONF [15], and there is a growing market of commercial OpenFlow-based SDN solutions [62].

On the other hand, T-SDN involves support of layer 2 and 3 and additional support for circuit-switched networks at layer 0 (optical) and 1 (SONET/SDH & OTN). Which entails significant challenges when compared with SDN solutions that focus only on layers 2 and 3. Therefore, the standardization process of T-SDN has been slower and remains an open issue. Nonetheless, there are early stage vendor solutions, mainly based on reuse of legacy technologies. Table 2.1 summarizes the characteristics of T-SDN and the challenges imposed by the optical infrastructure.

SDN programmability depends on the definition of common data plane abstractions and standard south-bound/north-bound interfaces [61]. At layer 2 and 3, accomplishing such features was relatively easy: indeed, in these layers the data plane abstractions can be defined upon well standardized packet headers. The exploitation of this advantage was the basis to deploy a common south bound interface like OpenFlow. Consequently, for packet-oriented networking manufacturers, it was simple to produce OpenFlow-enabled devices, that can be supported by commodity hardware, and a simple OpenFlow agent. Finally, OpenFlow agents could benefit from the well-consolidated techniques for packet classification based on standard layer 2 and layer 3 packet-fields.

Table 2.1: Transport SDN Characteristics

	Layer 2 and Layer 3	Layer 1 (OTN)	Layer 0 (optical)
Traffic model	Electronic packet-switching	Electronic TDM circuit-switching	Optical WDM circuit-switching
Data Plane operations	Packet header lookup, and packet operations (forwarding, encapsulation, pipeline processing, dropping statistics)	Operations over time slots (signal detection, transmission and switching). Performance monitoring	Operations over fibers, wavelengths (fixed and flexi-grid). Performance monitoring
Complexity	Low complexity: digital operations, based on packets headers	Relatively low complexity: digital operations, based on time slots	Highly complex: analogical operations, sensitive to physical layer constraints
Data Plane implementation	Homogeneous: based on standard protocols & specifications, vendor agnostic. Suitable for COTS devices	Homogeneous: based on standard protocols & specifications	Heterogeneous: vendor-specific features & configuration, administratively independent vendor islands
Data Plane abstraction	Easy to define standard abstractions	Relatively easy to define standard abstractions	Hard to define low level standard abstractions
Southbound interface	Standardized SBI e.g., OpenFlow	Non standard SBI, reuse of GMPLS and vendor-specific interfaces, multiple extensions proposed for OpenFlow (OpenFlow+)	
Control Plane	Standard OpenFlow-based control	Vendor-specific interface control, SDN/ GMPLS and ASON, OpenFlow-based control (not commercially available)	
Maturity	Standard commercial solutions and roll outs, based on OpenFlow	Non standard commercial solutions. Some OpenFlow standardization covered	Non standard commercial solutions

At the layer 1, composed mainly by OTN and its predecessor SONET/SDH technologies, it is as-well relatively easy to embrace SDN support [63]. Layer 1 OTN involves switching time slots in the electrical domain. Thus, all the signals are converted to the electrical domain, undergoing optical-to-electrical (O/E) and electrical-to-optical (E/O) conversions, in a hop-by-hop basis. OTN layer is well standardized by OIF, ITU and IETF, with standard compliant vendors' solutions.

The optical Layer 0, composed by fixed and flexi-grid (D)WDM technologies is the major challenge. We may say that the optical switching, that involves configuring OXCs at wavelengths and fibers, as in OTN is relatively easy. The optical switching capability allows to perform optical bypass. Thus, all optical paths i.e. lightpaths, are established to avoid O/E-E/O conversions in a hop-by-hop basis. However, the optical layer is transmission dependent, thus it is more complex than the layers above:

- The quality of signals in the optical layer is affected by photonic impairments such as chromatic and polarization mode dispersions, fiber nonlinearities and the Amplified Spontaneous Emission (ASE) noise [64].
- The optical systems are characterized by vendor-specific technologies and features like: switching constraints, power equalization, recovery mechanisms, and elastic transponder capabilities. For instance, among

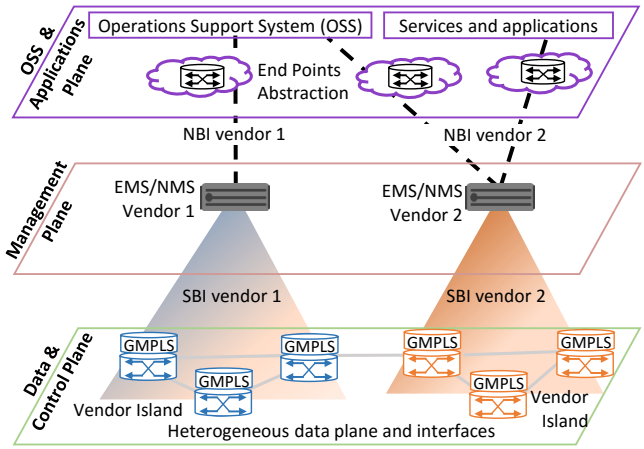

Figure 2.1: Legacy transport network architecture.

switching capabilities there is colored/colorless, directed/directionless, and blocking/contentionless [65, 66, 67, 68].

- Thus, differently from electrical infrastructure, in optical networks, transmission limitations translates into routing constraints for the logical layer of the lightpaths, i.e., transmission reach and wavelength continuity constraints.
- Therefore, at the optical layer not all the paths are feasible.
- Moreover, the optical networks continue to evolve, and present a gap between standardization and vendor implementations [60].

To cope with such complexity, optical network solutions rely on a vendor-specific management systems (e.g., NMS and EMS) that performs optical resource allocation, lightpath's reach computation, devices configuration, and quality of signals monitoring. As depicted in Fig. 2.1, current optical networks implement the GMPLS protocol suite as distributed control plane for dynamic path setup. The NMS together with GMPLS provide a "big switch" abstraction that hides the optical complexity and topology to the OSS and applications.

Historically, the optical network equipment providers have increased their solutions' competitive advantages by: introducing proprietary technologies with new features, and improving their management systems. This behavior led to heterogeneous data planes, with interoperability issues among diverse vendors' equipment. In consequence, the transport network of service providers is composed by administratively isolated vendor islands, each controlled by a centralized NMS. This heterogeneous scenario represents a big challenge to define common abstractions for T-SDN, and to gather detailed visibility and control over the multi-layer, multi-vendor, and multi-domain optical transport networks.

2.3.4 T-SDN classification

We can classify the T-SDN solutions by its control plane architecture in: monolithic T-SDN (SDON), hierarchical T-SDN (HT-SDN) and flat/mesh T-SDN (FT-SDN). Proposed by research efforts, the monolithic approach was the first step in the evolution of transport SDN . In literature we can find the term SDON (Software Defined Optical Networking) that refers to:

- Single SDN controller over a single optical domain: extensions that enable SDN at the optical layer 0 comprising software-defined transceivers, reconfigurable optical add/drop multiplexers (ROADMs) and cross-connects (OXCs) along with, extensions to SDN control plane and south bound interfaces (e.g., OpenFlow) [37][69].
- Single SDN controller over a multi-layer network that provides UCP of IP and Optical layers. With an SDN-enabled optical layer, SDON is able to exploit the benefits of a UCP for IP and optical layers [69].

However, the standardization bodies involved in SDN and transport networks (mainly ONF and OIF) agreed on a hierarchical architecture of controllers for transport SDN (HT-SDN) [9] [70]. The hierarchical architecture better suites the multi-domain nature of transport networks, where multiple domain controllers (SDN-based and legacy-based) are orchestrated either by a parent controller or by the transport network orchestrator. An SDON controller becomes a domain controller in the HT-SDN architecture.

A flat control plane architecture (FT-SDN) is composed by multiple domain controllers with a peer-to-peer coordination. Therefore, in opposite to HT-SDN that uses north bound and south bound interfaces for inter-controller communication, FT-SDN uses East/West interfaces for the peer-to-peer interaction between SDN controllers.

Flat control plane architectures were not the focus of T-SDN early stage development. The standardization of the east/west interfaces is far behind the achievements in north-bound/south-bound interfaces. For instance in [71] was proposed an inter domain protocol for the east/west interface between two EON domains. Peer-to-peer relations are expected to gain more interest for:

- Control plane clustering, which is supported by the latest version of ODL and by ONOS controllers, but is not well studied in literature.

- Inter provider coordination, where flat architectures are expected to be created among service providers [10].

In the following sections we provide a global view of the evolution of transport SDN, including research, standardization, and leading open source frameworks.

2.4 Research Efforts

In this section we start by describing the evolutionary path of research activity to enable SDN in transport networks. Based on the control plane architecture we classified the T-SDN evolution into two main groups: monolithic (SDON) and hierarchical (HT-SDN) architectures, which are presented in subsections 2.4.1 and 2.4.2, respectively. After enabling T-SDN, subsection 2.4.3 focus on the provision and support of virtual network (VN) services, a requirement for T-SDN [50] [70]. Subsection 2.4.3 presents a classification of research contributions on virtualization, virtual network embedding (VNE) algorithms in T-SDN scenario and T-SDN with network function virtualization solution (T-SDN-NFV). Tables 2.2, 2.3 and 2.4 summarize the time-line of the evolution of the previously mentioned subjects.

2.4.1 Monolithic architecture: SDON

The research activity in T-SDN control planes begins in 2009 at Stanford University with the so called Packet and Circuit Convergence (PAC.C) extensions to OpenFlow [63, 72], that proposed a monolithic control plane architecture.

2.4.1.1 Native OF support (PAC.C)

PAC.C theoretically and experimentally proved the viability and usefulness of migrating to SDN. PAC.C leads to convergence of packet and circuit domains into a flow-switched network with a unified control plane (UCP) that benefits from the visibility and control over IP and optical domains. It is based on a fully centralized architecture with a single SDN controller and native support of OpenFlow+ (OpenFlow extensions to manage circuit-flows) in network elements. The OpenFlow Circuit Switched Addendum v.03 [73] detailed a model to deploy circuit-flow tables into circuit-switching network elements at layer 1 and 0. PAC.C established a base-line approach for transport SDN.

Up to 2014, most of the research efforts shared a common target: to enable the optical data plane to be directly controlled by a single SDN/OpenFLow controller [38, 69]. We classify this solutions, characterized by the use of a single SDN controller to manage multi-layer transport networks as SDON. We can further classify the SDON solutions into:

2.4.1.2 Monolithic SDN/GMPLS interoperability models

It is a less disruptive approach than PAC.C, by reusing GMPLS as control plane of the optical domain [78, 1, 80, 121]. While extended OF -based control plane can: interface with GMPLS using the ASON-UNI, control the OF-enabled packet domains, and provide centralized network view and intelligence as shows Fig. 2.2.

Table 2.2: Time-line of research activities for enabling T-SDN with SDON

Year	Native OF	SDN-GMPLS	OF-Agent	Virtualization
2009	First OF+ L2 and L1 support [63, 72]			
2010	OF addedum [73]. OF+ L2, L1 and L0 support [73, 74, 75]			
2011	App-aware based TE and recovery [76, 77]	ASNO-UNI based [78, 1]	TL1 btw. OF-A and O-NE [2]. Dynamic RWA [79]	
2012		Parallel (RSVP-TE), (Overlay) OF-A based and integrated [80]. OF-A based [81]	Multi-technology UCP [82]. SNMP btw. OF-A and O-NE (ROADM), OF+ for optical constraints,[81]. Flexi-grid support [83]. Offload IA-R to PCE [84]	Optical Flow Visor (OFV) [85]
2013	Hybrid packet optical routers [86]		Multi-domain fixed and flexi-grid support [87]. Restoration [88, 89]. S-PCE to offload controller [90, 91]. OpenSlice BER monitoring [92, 89].	NETCONF/REST as SBI & NBI YANG for O-NE agents over EON [3]. GMPLS as a virtual control plane VONs[93]
2014			S-PCE [94, 95]	Equalization for ROADMs [96]. PCE based IA-RWA algorithm in [97]
2015				OSNR-aware modulation format reconfiguration [98]. Equalization, gain control and VON reconfiguration [99]

Table 2.3: Time-line of research activities for enabling T-SDN with HT-SDN

Year	OF-based	PCE-based	Hybrid (ABNO)	NBI-based using COP
2013	Hierarchy of NOX controllers [5]			
2014		ODL as Orch. and AS-PCE as ODC [6] network and IT orchestration [100].	First ABNO implementations [101, 8, 102]	
2015		SH-PCEs as Orch, S-PCE and S-PCE integrated into OpenFlow controller as ODCs over Flexi-grid domains [7]	REST APIs for ADVA ODC and PCEP for GMPLS domain [103]. OF-based ODCs [104]. Multi-technology including AS-PCE and BGP-LS for GMPLS domain [105]	RESTCONF -YANG [106]
2016				IT and network orchestration [107]

Orch.: orchestrator or parent controller - ODC: optical domain controller
AS-PCE: active stateful PCE - SH-PCE: stateful hierarchical PCE
COP: control orchestration protocol

Table 2.4: Timeline of research activities on T-SDN Virtualization

Year	SB Virtualization	Central Virtualization	NB Virtualization	VNE Algorithms	T-SDN-NFV
2012	Optical FlowVisor (SDON) [85],				
2013		NETCONF-based SDON [3]	Supported by OIF [70]	Multi-domain EON SDON [108, 109]	
2014	Comparison [110]. Hierarchical Virt. [111, 112]	Supported by ONF [9]			
2015	Hierarchical Virt. [113]		ABNO MD-Orch. [114]	VNE survivability [115]	SD-Orch. for multi-tenant TN [116]
2016			VN reconfiguration [117]		Ref. Arch. [118], Client's vT-SDN-C on the cloud [116]. MD-Orch. for multi-tenant TN [119], Mobile network [120]

VNE: Virtual Network Embedding - EON: Elastic Optical Network - vT-SDN-C: Virtual T-SDN Controller
SD-Orch.: Single optical Domain orchestration - MD-Orch.: Multiple optical Domains orchestration
TN: Transport Network

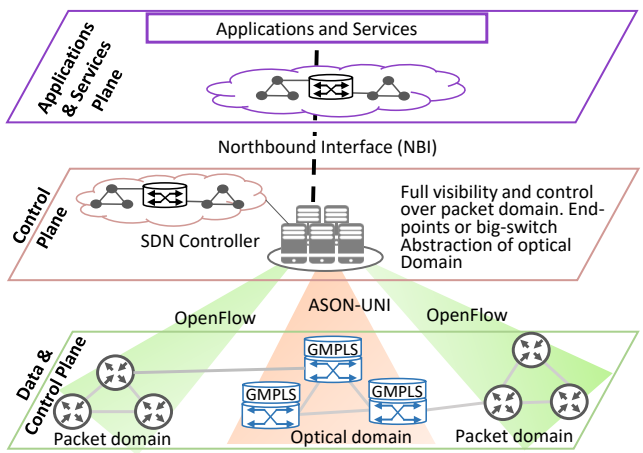

Figure 2.2: First proposal of SDN and GMPLS integration via ASON-UNI (SDN/UNI-GMPLS) [1]. The UNI offers a big switch abstraction with end points control and visibility.

2.4.1.3 OpenFlow Agent (OF-Ag) -based models

Great effort was made to develop OpenFlow agent (OF-Ag) solutions to convert legacy NEs into OF capable devices [2, 82, 81, 87, 122]. The first OF-Ag for optical NEs (depicted in Fig. 2.3) was proposed in [2] . [81] demonstrated the first SDON able to control commercial ROADMS. Multi-domain (packet switching, Optical Burst Switching, and Optical Circuit Switching) was achieve

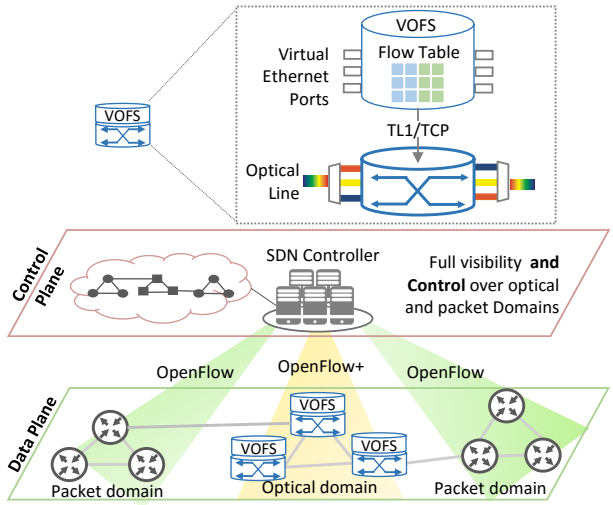

FIGURE 2.3: Architecture of the first pure OpenFlow agent-based model that provides full visibility and control of optical domains [2]. The agent (virtual OpenFlow switch; VOFS) interacts with the optical device management interface using TL1 and abstracts the optical node.

in [82, 122]. EON was first addressed by a control plane named OpenSlice [83, 92]. [87] demonstrated extensions for multi-domain packet over fixed and flexi-grid networking.

2.4.1.4 Transport network virtualization and NETCONF/YANG-based models

An alternative solution for the UCP of IP and optical domains that focus on providing network virtualization (NV) was presented in [85] with the optical flow visor.

A very interesting architecture that is recently gaining momentum is to employ NETCONF/REST as south bound interface for configuration of optical equipment and advertisement of its operational data. The first SDON NETCONF-based architecture (depicted in Fig. 2.4) was proposed by [3]. The agent maintains a YANG [4] database, that provides proper abstraction of optical NEs. The NETCONF-based controller have been used to prototype multiple control applications for the management of VONs over complex optical data planes such as global equalization algorithm for ROADMs [96], PCE based IA-RWA algorithm [97], EON modulation format reconfiguration according to OSNR thresholds [98], and the implementation of GMPLS as virtual control plane [93].

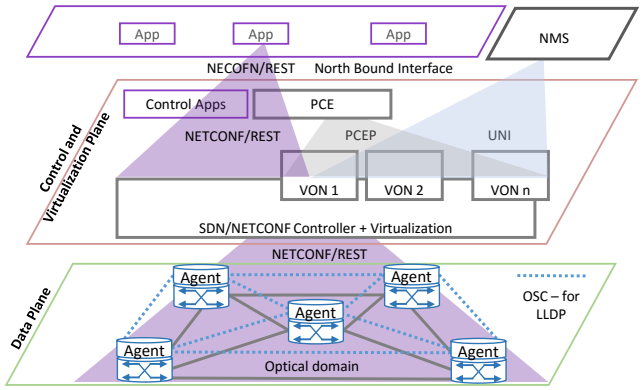

Figure 2.4: The NETCONF-based controller [3]. Agents on top of optical NEs are composed by YANG [4] database, that provides proper abstraction of optical NEs. YANG databases are populates with the link layer discovery protocol (LLDP) over an optical supervisory channel (OSC).

2.4.2 Hierarchical architecture: HT-SDN

After successful SDON proof-of-concepts, the transport SDN efforts have shifted the focus towards hierarchical control plane architectures. In this work, the hierarchical T-SDN architectures are classified as HT-SDN. The transport optical network is a complex system with heterogeneous domains from packets (Ethernet, MPLS and MPLS-TP) down to circuits (SDH/SONET, OTN and WDM). It is normally composed by vendor-specific islands each with proprietary and centralized management plane. Each domain runs with a combination of centralized and distributed proprietary control plane (e.g., ASON and GMPLS).

Fig. 2.5 depicts an example of the HT-SDN architecture. On top of the hierarchy, a parent controller or a transport network Orchestrator (TN-Orchestrator) application, interoperates with domain controllers to provision end-to-end and inter domain services. While at the domain level, specialized controllers are in charge of intra domain services. The hierarchical architecture increases scalability and allows better integration of the heterogeneous domains. Several standardization bodies led by ONF [9] and OIF [70] support such hierarchical architecture. Through definition of proper abstractions and interfaces the HT-SDN architecture is able to control multiple vendor islands based on standard and proprietary technologies and protocols.

The following subsections describe the research activities for enabling T-SDN using HT-SDN. As summarized by Table 2.3, we propose a classification of HT-SDN based on the protocols used among the hierarchy of controllers:

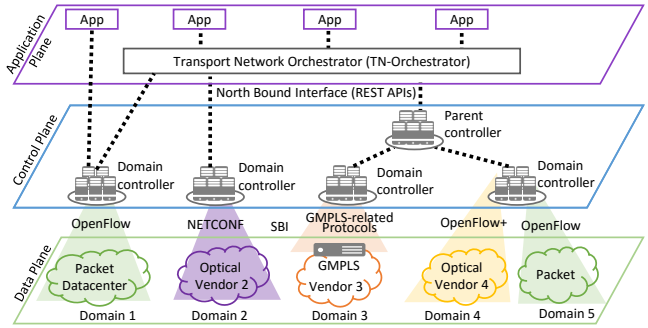

FIGURE 2.5: Hierarchical T-SDN (HT-SDN) architecture.

2.4.2.1 HT-SDN based on OpenFlow

Following an ONF-like approach [5] presented an HT-SDN using a hierarchy of NOX [123] controllers, that employ OF as interface between domain and parent controllers as depicted in Fig. 2.6.

2.4.2.2 HT-SDN based on AS-PCE (using OpenDaylight)

In [6] OpenDaylight (ODL) [31] was used as parent controller of GMPLS and packet domains as Fig. 2.7 illustrates. An active stateful PCE (AS-PCE) was used as domain controller over a GMPLS domain. The packet switched domains are directly controlled by OpenDaylight using OpenFlow. The AS-PCE serves as domain controller, that provides a hardware abstraction layer with full visibility over the GMPLS domain. The PCEP plug-in provided by ODL was extended to support active stateful PCE [124]. Orchestration applications were deployed using the REST APIs offered by ODL for topology acquisition and

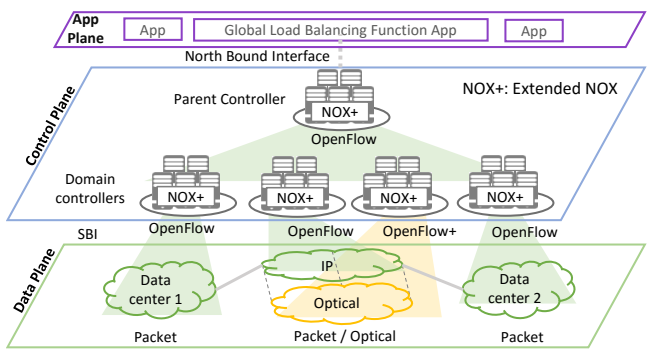

FIGURE 2.6: All-OpenFlow HT-SDN architecture [5].

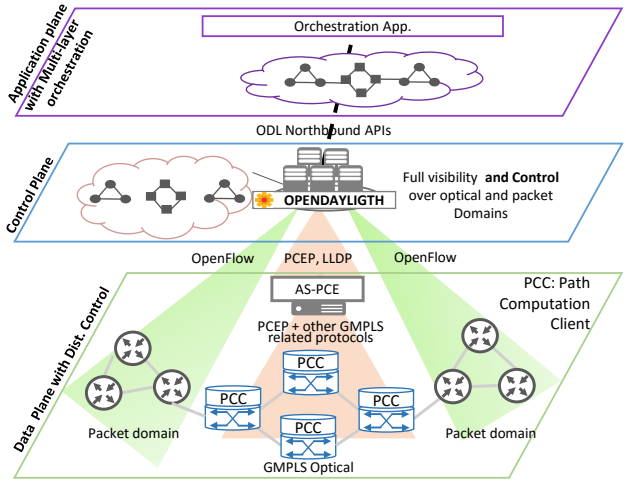

Figure 2.7: Hierarchical integration of SDN and GMPLS control planes over packet and optical domains using OpenDaylight and AS-PCE [6].

end-to-end path computation across the multiple domains. The architecture of Fig. 2.7 was used to demonstrate orchestration of network and IT resources for inter and intra datacenter dynamic control [100].

2.4.2.3 HT-SDN based on SH-PCE

[7] proposed to use stateful hierarchical PCE (SH-PCE) for HT-SDN as presented in Fig. 2.8. A parent/orchestrator PCE coordinates inter domain path computation over three child stateful PCEs (c-S-PCE). One c-S-PCE directly governs a flexi-grid DWDM network. The other two c-S-PCEs are integrated inside OpenFlow controllers to support flexi-grid DWDM networks. The main contribution of this work is to extend H-PCE with stateful capabilities, and the integration of child S-PCE with OpenFlow controllers.

2.4.2.4 HT-SDN based on REST APIs and PCEP (using ABNO)

The IETF SDN research group proposed a modular and multi-domain SDN orchestration architecture called the PCE-Based Architecture for Application-Based Network Operations (ABNO) RFC 7491 [11] (described in subsection 2.6.3).

The first experimental demonstration [101, 103] and the main efforts in ABNO-based HT-SDN have been developed in the framework of the European projects STRAUSS and IDEALIST. [101, 103] demonstrated automatic provisioning of IP connections (Juniper routers) across two optical domains, one with an emulated GMPLS control plane and other with an ADVA SDN/OpenFLow

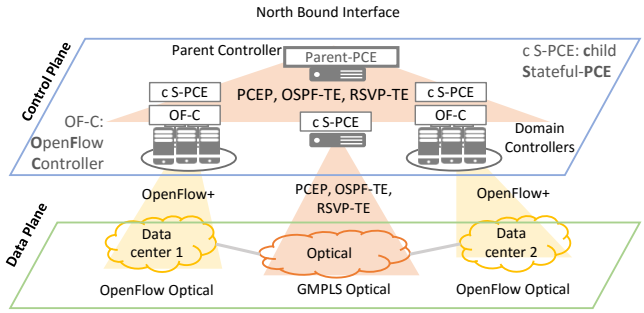

FIGURE 2.8: HT-SDN architecture based stateful hierarchical PCE (SH-PCE) [7].

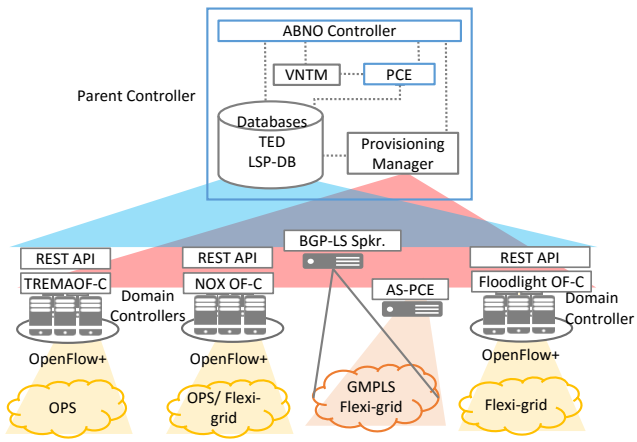

FIGURE 2.9: ABNO-based orchestration over heterogeneous OPS and OCS domains [8].

controller based on Floodlight [125]. The optical layer was configured using: the REST API provided by ADVA SDN/OpenFlow controller, and the PCEP for the GMPLS-controlled domain.

HT-SDN over multi-domain and multi-technology of SDN/OpenFlow-based domains using ABNO was presented for the first time in [102, 104].

In [8, 105] orchestration of end-to-end services over two SDN/OpenFlow controlled OPS domains, two SDN/OpenFLow controlled OPS/OCS domains, and a GMPLS/PCE controlled OCS domain with AS-PCE and BGP-LS speaker. Fig. 2.9 shows the international testbed including OpenFlow enabled OPS, OPS/flexi-grid and flexi-grid domains, and one GMPLS controlled flexi-grid domain.

2.4.2.5 HT-SDN based on the Control Orchestration Protocol (COP)

The control orchestration protocol (COP) is a solution from the STRAUSS European project to allow interoperability among heterogeneous multi-domain, multi-technology transport networks [106, 107]. COP is intended for the north bound interface of diverse control plane technologies. It provides REST APIs using RESTCONF. Technology-specific data models are defined using YANG.

2.4.3 Virtualization

Thanks to the multiple abstraction layers provided by SDN, T-SDN enables efficient and flexible virtualization of transport networks. A virtual network (VN) is a logical topology composed by virtual nodes and virtual links mapped into a physical infrastructure. Multiple VNs share a common networking infrastructure, each with distinct forwarding logic, and isolated from each other. In the context of SDN, an instance of a virtual network is commonly called slice [126]. Each slice can be separately managed by a guest or internal SDN controller.

We refer to the hypervisor as the virtualization platform or layer that enables distinct slices to share a common networking infrastructure. The hypervisor introduces another abstraction layer into SDN architecture to allows the creation and management of network slices. Moreover, SDN allows to jointly optimize the virtual embedding of network and computation infrastructure. Support provision of VN services is a requirement for T-SDN [50] [70]. In the following subsections we introduce a classification of virtualization architectures for T-SDN, discuss algorithms for VN embedding in T-SDN, and present implementation strategies for network function virtualization (NFV) in T-SDN called T-SDN-NFV. Table 2.4 presents a timeline of the virtualization architectures, virtual network embedding (VNE) algorithms in T-SDN scenario and the T-SDN-NFV.

2.4.3.1 Virtualization architectures for T-SDN

As in several issues related to T-SDN, there is no consensus or stable standardization to provide VNs in transport networks. We present a classification of VN architecture solutions for T-SDN based on the location of the virtualization platform or layer.

- **Southbound Virtualization**

  A hypervisor layer is placed between the data plane and the control plane. Tenant controllers are deployed over the virtual networks provided by the virtualization platform.

  The Optical FlowVisor [85] was the first proposal for virtualization of transport networks. The Optical FlowVisor was proposed for the mono-

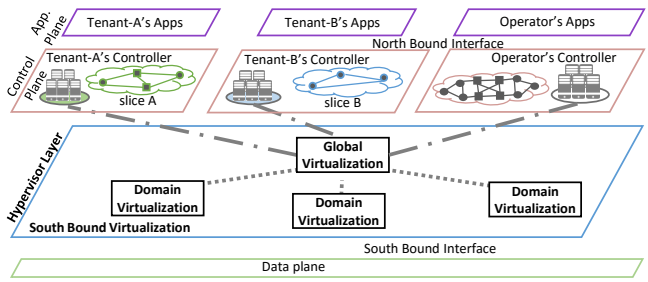

FIGURE 2.10: South bound virtualization architecture for the multi-domain scenario.

lithic T-SDN controller architecture, i.e. SDON discussed in subsection 2.4.1.4. It is based on FlowVisor [126], one of the first platforms for network slicing in SDN. In [110] was also considered the south bound virtualization for T-SDN, using three degrees of topology abstraction: single node (big switch), full topology (no abstraction), and abstract link model (provides abstraction between the previous models). Based on the analysis done in [110], the abstract link model presents the best trade off between manageability and complexity towards the tenant controllers.

In order to apply south bound virtualization in multi-domain transport networks, a hypervisor layer, composed by a hierarchy of virtualization platforms needs to be implemented. Fig. 2.10 depicts the hierarchical architecture of south bound virtualization for multi-domain scenarios. In [111, 113, 112], technology-specific virtualization platforms were placed on top of heterogeneous domains, and a global or parent virtualization controller creates the hierarchical virtualization layer. The global virtualization has some functionalities of an SDN controller: gather a global network view, network resource assignment, VN request handling and VNs construction.

- **Central Virtualization**

  The virtualization platform is placed inside the SDN controller. It uses internal controller interfaces to gather and control an abstract view of the network. Using central virtualization, the SDN controller provides internal NBI for creation and management of slices.

  The central virtualization was first used in [3], using a NETCONF-based controller, in the context of SDON architecture, discussed in subsection 2.4.1.4. The NETCONF-based controller was able to support heterogeneous control planes for the network slices.

  Fig. 2.11 presents the general architecture of central virtualization for multi-domain networks, where virtualization functionalities are placed

at the top of the hierarchy of controllers, i.e., in the parent or global controller of HT-SDN. By exploiting the multi-domain capabilities of the parent controller, central virtualization is easier to deploy in multi-domain scenarios than south bound virtualization.

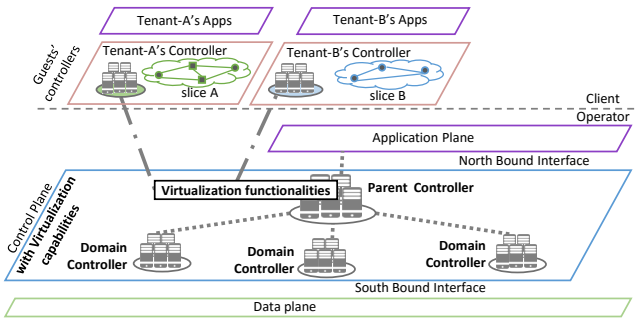

Figure 2.11: Central virtualization architecture for the multi-domain scenario.

Central virtualization is supported by ONF. In the ONF OpenFlow-enabled T-SDN reference architecture [9], the virtualization platform is considered as functionalities of the Global Controller. ONF even defined a Control Virtual Network Interface (CVNI), as the interface used between controllers, including the one between global controller and virtual tenant controllers (see section 2.6.1).

- **Northbound Virtualization**

  The hypervisor is placed above the control plane, or even on top of the transport network orchestrator, as Fig. 2.12 depicts. North bound virtualization, exploits the multi-domain capabilities and global abstracted view provided by the control plane or orchestrator. Thus it can be also called abstracted virtualization.

  North bound virtualization is supported by OIF [70]. For OIF the orchestrator must control slicing of transport network infrastructure. Moreover, a higher level orchestrator with management capabilities over network and IT resources must provide virtualization for transport networks (e.g., NFV) and datacenter (e.g., virtual machines).

  In [114], was developed a north bound network hypervisor, that exploits the APIs provided by an ABNO-based multi-domain network orchestrator. The network hypervisor, allows OpenFlow based guests' controllers to manage their own slice of the network. The ABNO orchestrator is in charge of guaranteeing end-to-end QoS for each slice, over multi-technology and multi-domain networks.

  In [117], was demonstrated a north bound virtualization that provides dynamic VNs which are able to react upon congestion and failures. A

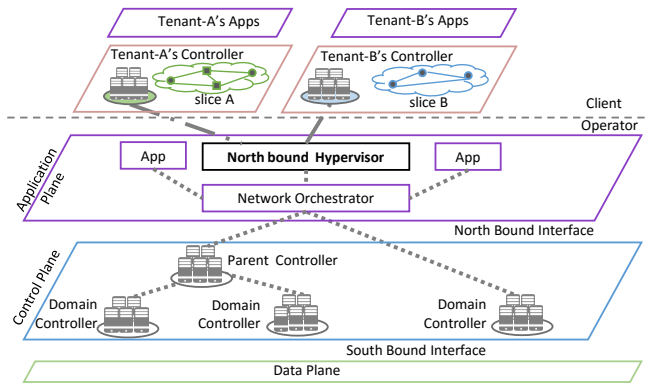

FIGURE 2.12: North bound virtualization architecture for the multi-domain scenario.

north bound virtualization platform exploits ABNO orchestrator capabilities for re-planning and recovery mechanisms, and changes applied bellow the orchestrator are transparent for the VNs.

2.4.3.2 T-SDN compatible VN embedding algorithms

In [108] was studied the virtual infrastructure embedding (VIE) problem for multi-domain flexi-grid networks controlled by a monolithic SDON architecture. The authors of [108] proposed a virtual link embedding algorithm to maximize the number of VONs embedded in the physical substrate, while taking into account transmission reachability and wavelength continuity constraints. The virtual link embedding in flexi-grid network involves assignment of routing, modulation format and spectrum. The research in [109], extended [108], to perform both virtual link and node embedding of network and computing resources. Later in [115], was introduced the survivability against single node or link failure to the virtual embedding problem for T-SDN.

2.4.3.3 T-SDN and Network Function Virtualization (NFV)

Network Function Virtualization (NFV) is a network architecture paradigm that leverages virtualization techniques to dynamically deliver Virtualized Network Functions (VNF)s. VNFs are software implementations of physical network functions (PNF)s, including data, control and management functionalities, which are necessary to run a network. NFV decouples software from hardware, thus, VNFs can be dynamically instantiated into a cloud computing environments using Commercial Off-The-Shelf (COTS) hardware, instead of running into function and vendor-specific hardware.

NFV and SDN are two networking technologies that shift the paradigm of doing networks. SDN decouples control from data plane, while NFV decouples software from hardware. However, in this work we have presented T-SDN architectures without NFV. In the same way, NFV can be implemented without SDN technologies. In fact, the first attempt to introduce NFV in transport networks did not use an SDN control plane, but a GMPLS/PCE control plane [127]. NFV was used to virtualize the PCE as a VNF. A PCE NFV Orchestrator creates and releases virtual PCEs (vPCE)s dynamically, adapting to demand variations of path computation requests. BGP-LS was not enabled to acquire the topology, thus all vPCEs share an static topology. A path computation entity must first consult the IP of the vPCE to a PCE DNS, which is responsible of vPCE load balancing.

However, providers are willing to use both technologies to boost flexibility, speed up deployment times and reduce costs. For instance, Verizon published the SDN-NFV Reference Architecture [118], based and co-authored with multiple vendors.

In [116] and [119], the NFV architecture together with the virtualization architectures of T-SDN were used to virtualize client's SDN controllers into the cloud. By doing so, virtual T-SDN networks can be provisioned dynamically on demand, the controller can be relocated due to changes on the demands or to recover from a disaster. For instance subsection 2.4.3.1 each client's controller runs in client-specific facility using dedicated hardware.

End-to-end orchestration (EEO) is one of the main components of the SDN-NFV architecture [118]. In the context of T-SDN-NFV architecture, the EEO should be extended to support orchestration of NFV, SDN services and end-to-end network services.

The T-SDN and NFV orchestrator for multi-tenant transport networks was first demonstrated in [116], over single domain optical network. This orchestrator exploits NFV and south bound virtualization provided by the optical network hypervisor of [110]. The average provisioning time of a virtual T-SDN, reported in [116], is less than 2 minutes, and involves multiple requests form the SDN NFV orchestrator: 1) creation of virtual SDN controller to a VNF manager, 2) flows setup between virtual controller, optical network hypervisor and client's network operation center, 3) creation of VON to the optical network hypervisor.

To demonstrate virtualization of tenant's controllers over heterogeneous multi-domain transport networks, the authors of [119] employed the north bound virtualization approach together with NFV. The architecture exploits the multi-domain capabilities of the ABNO-based orchestrator [114], and the north bound virtualization provided by the multi-domain network hypervisor [117].

In [120], the SDN NFV architecture of [119], was extended to the mobile network. Radio Access Network (RAN) are connected to datacenter facilities through the backhaul network. The backhaul, is composed by multiple domains with diverse transport network technologies. In [120], was demonstrated the

virtualization of the backhaul network, with VNF-like SDN controllers and evolved packet core. Such virtualization can be greatly exploited in the mobile network to cope with the traffic demand variations.

In early 2016, the Linux Foundation formed the OPEN-Orchestrator Project (OPEN-O) to develop the first open source software framework and orchestrator agile operations of SDN and NFV. As an open source orchestration framework, OPEN-O will integrate open networking technologies and enable carriers to quickly and cost-effectively implement SDN and NFV through open source code development [128]. Open-O aims for multi-vendor integration, service innovation and improve agility across network operations. As in the case of OpenDaylight, OPEN-O will accelerate the implementation of orchestration engines.

2.5 Other Research Developments

2.5.1 Protection and Restoration

In transport networks, failures may lead to huge amount of data loss. Thus, multiple protection and restoration schemes have been proposed in literature. In a protection scheme, the connection is provided with a primary and a protection path. The primary is used during normal operation. The protection path is used only after the primary path is affected by a failure. In a restoration scheme, the network reacts after failures to re-provision the disrupted connections. Restoration schemes avoid provisioning high amounts of bandwidth for protection. However, it does so with a penalty in longer recovery times, and lower guarantee to re-provision affected connections.

T-SDN has properties to improve restoration and protection schemes, for instance: 1) the centralized nature that allows to set up path faster by sending flow set up messages in parallel to all the nodes of a path, and increases the dynamicity of the network. 2) The network wide view gathered at the control and application plane that allows to implement optimization algorithms. 3) The multi-layer and multi-domain capabilities that can unify the protection and restoration schemes across multiple layers and domains [129, 130]. In the following, this subsection presents a list of proposals on protection and restoration for T-SDN.

- In [131] and [132] the authors presented an SDN/OpenFlow-based restoration mechanism for EONs that takes into account the physical impairments to improve the efficiency of the controller to find feasible restoration paths. The proposed extensions to the OpenFlow included those in support of EON from [92] and the definition of a new message to support the alarms (notification of link failures) in the network. The solution presented in [132] included a mechanism to determine specific single-point of failure by combining the information of the network topology and the current established lightpaths in the network. After the failure point is determined, the

controller first deletes the flow entries of the working path and then set up the new path obtained with a two-phase restoration routing, spectrum, and modulation format assignment (RSMA) algorithm. The proposed restoration function was tested in the GENI (Global Environment for Network Innovations) [133] testbed using an emulated data plane with 14 BV-WXC and a NOX-based [123] controller.

- In flexi-grid networks, the spectrum selective switches (SSS) requires longer configuration times (25 milliseconds) than the OXCs of WSON. Thus, in [134], authors demonstrated that the centralized nature of SDN can improve the recovery time of flexi-grid networks, and that outperforms the GMPLS/PCE control plane restoration schemes. Giorgetti et al. prototyped, by means of simulation, an SDN-based scheme to minimize the number of node reconfigurations and contentions during recovery of flexi-grid networks. Their scheme minimizes the overall recovery time and restoration blocking probability by bundling the reconfiguration instructions for all the affected lightpaths. In [135], authors extended the simulation scenarios, showing that the contentions (spectrum and node configuration) among different recovery signaling sessions, that are very likely to happen in a distributed control plane like GMPLS, increase the recovery time.

- A multipath protection scheme for OpenFlow-based flexi-grid networks was demonstrated in [136]. The authors extended the cross-connection table proposed in [83], by adding a field to specify the Type of signal using integer positive numbers. The Type field was used to distinguish between the primary path (Type = 0) and the set of n disjoint path for protection (Type = $n > 0$). The smaller field Type is, the higher the priority of the path is. The multipath resource allocation is done upon a new connection arrives using bandwidth squeezed protection. Upon failures, the controller determines the disrupted paths. Then, it sends FLOW_MOD messages to delete flow entries at the source node of all affected paths. Thus, each connection is forwarded along the remained protection path with highest priority. Results on a simulated environment demonstrated a reduction on blocking probability when using multipath protection against no protection. A drawback of this scheme is that it provisions a large amount of bandwidth for the multiple protection path.

- The authors of [137] proposed a Backup Reprovisioning with Partial Protection (BRPP) scheme for WDM networks, for disaster survivability. Thanks to SDN, the BRPP can use a mixture of logical protection and physical restoration. After every network state change, BRPP runs a global backup reprovisioning heuristic in an abstracted view of the network (at application plane) to calculate the protection paths. Only after failures, BRPP establishes the protection path in the data plane. Thus, avoiding extra delays due to contentions in the data plane. BRPP

considers resource reallocation of disrupted and non-disrupted paths based on degraded service tolerance, to reduce blocking probability of the backups.

2.5.2 Segment Routing

Segment Routing (SR), also called Source Packet Routing in Networking (SPRING), was proposed by Cisco in 2013 and it is based on the source routing paradigm [138]. SR provides traffic engineering (TE) solutions, while addressing several control plane drawbacks of legacy IP/MPLS networks e.g., improve scalability, simplicity, and ease of operation. SR is being standardized by the IETF SPRING working group [138]. In SR, segment identifiers (SID)s are labels, encoded in 32 bits MPLS labels, that represent intermediate path points. A path is specified at the source node using an ordered list of SIDs, compatible with an MPLS label stack.

SR was built for centralized control plane architectures, for instance SDN. SR offers the possibility to combine the advantages of distributed (e.g., MPLS and GMPLS) and centralized (PCE and SDN) control planes. Moreover, using SDN-based SR, the controller reduces the signaling, as it does not need to configure every node belonging to a flow. The controller need just to send configuration packets to the source node of the flow.

It is not surprising that there are works that already target SDN-based SR for multi-layer and multi-domain networks. Sgambelluri, *et al* [139] proposed the first SDN/OpenFlow-based SR implementation for multi-layer packet-optical network. In [139], was demonstrated dynamic packet rerouting with optical bypass capabilities. Later, Sgamberulli, *et al* [140], extended their work for multi-domain and multi-layer scenarios, using a mesh control plane architecture. Using non-standard east/west interfaces, the authors proposed a methodology to exchange intra-domain SID information. In chapter 6 we present the implementation of a hierarchical SDN control plane, using standard north bound (APIs) and south bound (BGP-LS and PCEP), to allow orchestration of multi-domain SR networks [141].

2.5.3 Emulation

Among the SDN network emulation platforms, Mininet is the most popular open source solution [142] [143]. Mininet uses lightweight-virtualization mechanisms such as processes and virtual Ethernet pairs in network namespaces. The lightweight-virtualization based emulation allows to test large network instances in a laptop, which is not possible using a full-system emulation that uses one virtual machine per network element.

However, Mininet or any other open available SDN emulators do not support optical network elements.

To the best of our knowledge, SONEP (Software-Defined optical Network emulation platform) was the first SDN optical network emulator [144]. SONEP

is a container based emulator composed by: virtual OTS from Infinera, virtual links (WDM and ethernet), virtual hosts and OpenFlow switches. Even though it is a promising solution for fast prototyping of T-SDN, SONEP is not available and cannot be used by the T-SDN research community.

More recently, LINC-OE (LINC Switch for optical emulation) was developed by Infoblox/Flow-Forwarding community in collaboration with ON.Lab for the inclusion into the ONOS packet/optical convergence use case [145]. LINC is a software switch that supports OpenFlow and OF-config [146]. When the LINC switch is configured with a backend to emulate ROADMs, it becomes into an optical emulator, where each ROADM run as logical switch within LINC-Switch container [147] LINC-OE supports OpenFlow, and uses proprietary extensions based on the "experimental" capability, in line (but not interoperable) with the ones presented in [49]. LINC-OE allows to use Mininet for configuration, and support failures at links, ports and ROADMs.

A public repository from Telefonica I+D provides a Java based Emulator of a Transport Node (L1/L0) with GMPLS control plane [148].

2.6 T-SDN Standardization Efforts

Multiple standardization bodies are working on defining standards for T-SDN including ONF, OIF, IETF and ITU-T. Until now ONF and IETF are the main organization for T-SDN standardization. The main efforts are based on OpenFlow (ONF) and GMPLS (IETF), and recently the work on the transport APIs is gaining momentum. However there is a long run to have stable standards on T-SDN. Table 2.5 summarizes the main efforts on standardization of T-SDN and 2.6 specifically presents the standardization of transport networks APIs.

2.6.1 Open Networking Foundation (ONF)

The ONF [15] is a young organization (2011) dedicated to the promotion and adoption of SDN and OpenFlow through open standards development.

In 2013 ONF chartered the Optical Transport Working Group (OTWG) (renamed as Open Transport WG) to develop OF extension to support optical transport networks [15].

Fig. 2.13 depicts the ONF reference architecture [9], which is based on a hierarchy of controllers that communicates between each other by means of the OpenFlow. As presented in Fig. 2.13 there are two OF-based south bound interfaces:

- Control Data Plane Interface (CDPI): like an SBI it is used between controllers and network elements.
- Control Virtual Network Interface (CVNI): used for interaction among domain and virtual (tenants) controllers.

At the end of 2013 the ONF published the Openflow Switch Specification v1.4.0 [48] that introduced for the first time support for optical ports. A set

Table 2.5: Timeline of Standardization on T-SDN architecture, control plane and south bound interfaces

Year	ONF	OIF	IETF
2013	OF v1.4.0 [48] first time support for optical ports	T-SDN Requirements [70]	
2014	OF-based HT-SDN reference arch. [9]	Global demo with multi-domain, multi-vendor and multi-carrier scenario composed by L2 and L1 (OTN) swithces [17]. APIs	
2015	T-SDN Recommendations that included L0 and L1 operations [49]	HT-SDN arch. with ASON-based REST APIs [10]	RFCs: PCE [59], NETCONF [149], YANG [4], ABNO [11], VNTM [150], PCEP [151], ALTO [152]
2016			Work in progress: S-PCE [153], AS-PCE [124]

Table 2.6: Timeline of Transport API standardization efforts

	Northbound			Southbound
Year	ONF	OIF	IETF	OPEN-ROADM
2015	ONF/ ITU-T CIM [154, 155]	ASON-based REST/JSON APIs [10]	NETCONF [149], YANG [4], JSON enconding with YANG [156]	
2016	TAPI requirements [157]		TE Topo. [158], [159]. VNT [160].TN [161].PCEP [162].[163]. OTN [163]WSON [164], EON [165]	Disaggregated ROADM with NETCONF APIs [166], YANG models [167]

CIM: Common Information Model - TAPI: transport API - TE Topo.: Traffic Engineered Topology ASON: Automatically Switched Optical Network - VNT: Virtual Network Topology - OTN: Optical Transport Network - WSON: Wavelength Switched Optical Network - EON: Elastic Optical Network

of optical port properties allow the configuration and monitoring of frequency and power of transmitted and received optical signals. However, the optical extensions presented in [48] are very limited.

Based on ONF requirements analysis [50] and OIF carrier requirements [70], the ONF-OTWG published a set of recommendation for T-SDN [49]:

- Match and Actions extensions:
  - Match extensions for identifying signals at layer 0, using attributes of an OCh: Grid, Channel Spacing, center frequency, channel mask.
  - Match extensions for identifying signals at layer 1, using attributes of an ODUj/k: ODU type, ODU Tributary Slot, ODU Tributary Port Number signals.
  - No Action extension: the *SET_FIELD* mechanism is used to specify the attributes of the egress signals, thus without incurring into OpenFlow Action extension.
- Port attributes extension to identify port types at L0 and L1 of OTN standard interfaces.
- Adjacency discovery for OTN transport networks based on the in-band exchange of identifier information as defined in ITU-T G.7714.1 [168].

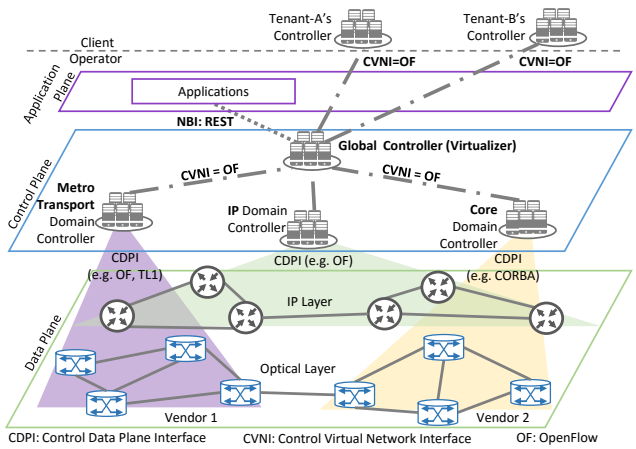

Figure 2.13: Reference Hierarchical Control Architecture proposed by ONF [9].

Future extensions considered by the ONF-OTGW includes: OAM/Monitoring of optical network links, connection protection, multilayer connections and the use of OpenFlow among controllers in the CVNI (Control Virtual Network Interface).

2.6.2 Optical Internetworking Forum (OIF)

The OIF is an organization dedicated to facilitate and improve: interoperability, cost-efficiency and robustness of optical internetworks.[2]

In 2013 the OIF summarized requirements for deployment of T-SDN [70]. Fig. 2.14 depicts the reference architecture envisioned by the OIF, based on the HT-SDN architecture of SDN and on the ITU-T ASON control plane model. The main difference between the OIF and ONF reference architectures is that, for OIF OpenFlow is not the main protocol to use. In [10] are identified already existing protocols that can be reused at the SBI and NBI of transport SDN. For instance, the SBI of domain controllers should provide a variety of protocols to interact with the infrastructure layer (or data plane). In a transport network, the infrastructure layer may include brownfield domains that use a distributed control plane (GMPLS or ASON) or a centralized network management systems. Thus, the provisioning of diverse SBI protocols allows to interact with: network elements, distributed control planes, and centralized network management systems.

In 2014 OIF and ONF joint efforts to test a prototype transport SDN technologies in a real-world multi-domain, multi-vendor and multi-carrier scenario,

[2]Optical internetworks are data networks composed of routers and data switches interconnected by optical networking elements.

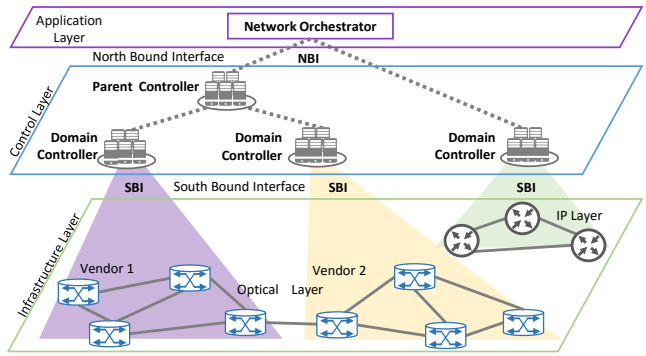

FIGURE 2.14: HT-SDN architecture proposed by OIF [10].

composed by Ethernet and OTN (ODU and OCH) switches [17]. The prototype implementation was based on the reference architecture proposed by ONF [9] (see Fig. 2.13).

Apart form Openflow, the domain controllers supported vendor-specific SBIs. The demonstration proved to be an effective approach to have domain controllers that provide diverse SBIs towards heterogeneous optical elements. The OIF was the first to proposed a reference for standard transport APIs (REST and JSON -based), as the key to foster interoperability in the transport network multi-vendor scenario [10] (see subsection 2.7.2 for more details).

2.6.3 IETF

The Internet Engineering Task Force (IETF) is an open international community of network designers, operators, vendors, and researchers concerned with the evolution of Internet architecture and its smooth operation.

There are IETF working groups to cover several areas of SDN, for instance:

2.6.3.1 GMPLS/PCE and SDN Interoperability

GMPLS is the most popular control plane for optical transport networks. For an overview of the interworking between GMPLS and OpenFlow we refer the reader to [42]. The PCE is the key element for centralization of path computation tasks over a GMPLS domain, while PCEP and BGP (BGP-LS) allows GMPLS and SDN interoperability. The SDN controller can use PCEP for provisioning of LSPs, and BGP-LS to get topology visibility.

2.6.3.2 Active Stateful PCE (AS-PCE) & PCE protocol (PCEP)

The PCE described in [59] was conceived to decouple and centralize the path computation from the distributed control plane of MPLS and GMPLS. Ac-

cording to RFC 5440 [151], the PCE can only compute a path upon receiving a request from a Path Computation Client (PCC) or other PCE, which is not compatible with the SDN paradigm. IETF-PCE working group is being developing PCEP extensions to boosts control, visibility and scalability of the PCE [153][124]. Such work is contributing to standardize the interoperability between SDN and GMPLS. The proposed PCEP extensions added three main capabilities to the PCE: stateful, active and hierarchical [153, 124].

2.6.3.3 North-Bound Distribution of Link-State and TE Information using BGP messages (BGP-LS)

: The extensions to the border gateway protocol (BGP) proposed in [169] allow a PCE (or a network controller) to access the TE databases (TED) of IGP area(s) or autonomous system(s). The BGP-LS allows to collect, filter (based on policies) and distribute with a PCE the LSDB and TED from IGP areas.

2.6.3.4 Network Configuration Protocol (NETCONF)

The Network Configuration Protocol (NETCONF) RFC 6241 [149] is a standardized network management protocol that provides mechanisms to install, manipulate, and delete the configuration of network devices. NETCONF allows a network element to expose a standard application programming interface (API), which is very suitable for SDN environments. NETCONF uses an Extensible Markup Language (XML)-based data encoding for configuration and protocol messages.

2.6.3.5 YANG modeling language

YANG RFC 6020 [4] was conceived as the data modeling language for NETCONF protocol, but is becoming also relevant for REST-based interfaces using JSON encoding instead of XML. Thus, YANG represents the basis of SDN programmatic APIs implementation (see section 2.7.1).

2.6.3.6 Application Based Network Operation (ABNO)

ABN RFC 7491 [11] represents an attempt to standardize the building blocks and internal interfaces of T-SDN controller reusing IETF components. ABNO was conceive to allow the interoperability between legacy (e.g., IP/MPLS, GMPLS) and OpenFlow domains. It avoids vendor lock-in and provides support for the NMS and Operations Support System (OSS).

Fig. 2.15 illustrates the generic ABNO architecture as proposed by RFC 7491 [11]. As depicted in Fig. 2.15 the NMS, OSS and the set of applications that are called the Application Service Coordinator (ASC), communicates with the north bound interface (NBI) of the ABNO framework. It is worth noting that there is no standard definition for the NBI of the ABNO.

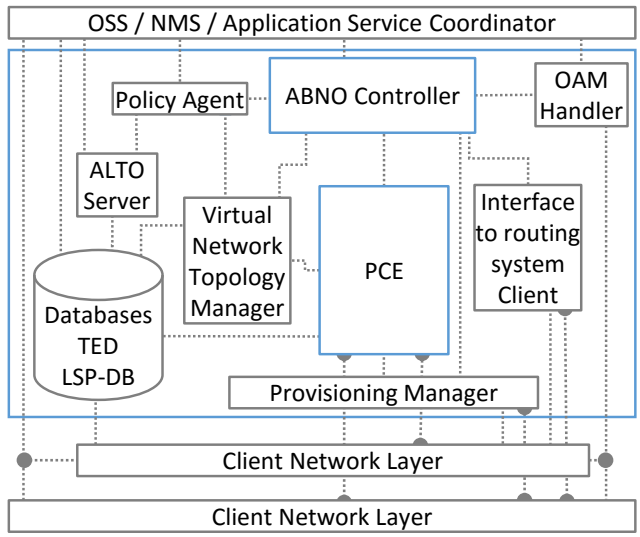

FIGURE 2.15: Generic ABNO architecture from RFC 7491 [11].

The ABNO controller is the central component, it provides the interfaces to the NMS, OSS and applications towards the network, and coordinates the workflow among other ABNO blocks in alignment with NMS, OSS, and applications requirements and the current network conditions.

There are two main databases. The Traffic Engineering Database (TED) stores the topology information and can alternatively include capacity and status of the elements. The Label Switch Path Database (LSP-DB) that includes the paths and resources assigned to LSPs, that are currently or to be established in the network.

The PCE described in [59] handles constrained path computation over a network graph provided by the TED, and it is one of the main components of the ABNO framework.

The Virtual Topology Network Manager (VTNM) is defined in RFC 5212 [150], and it is in charge of multi-layer path provisioning.

The Provisioning Manager (PM) provides the appropriate interfaces for the establishment of LSPs in the network. In a hierarchy of controllers, the PM is able to interact with the control plane of the network domains (domain controllers e.g., GMPLS, AS-PCE, SDN) using the NBI of such control planes (PCEP [151], NETCONF RFC6241 [149], REST APIs). Additionally in a UCP approach, the PM is able to directly use the proper interfaces (ForCES RFC5810 [170], NETCONF RFC6241, OpenFlow) to interact with individual network devices.

The ABNO framework includes the Interface to the Routing System (I2RS), that is a work in progress described in draft [171].

The Application-Layer Traffic Optimization (ALTO) RFC 7285 [152] server can be also part of the ABNO framework, it provides abstract representations of the network to applications on top of ABNO.

2.6.4 ITU-T

ITU-T is the standardization sector of International Telecommunication Union (ITU) [168]. The ITU-T Joint Coordination Activity on SDN (JCA-SDN) published a roadmap to keep an up-to-date information on all standardization activities on SDN [172].

The ITU-T Study Group 15 (SG15) is studying Transport aspects of SDN in close alignment with the ONF. The SG15 is working on two drafts Recommendations: "Architecture for SDN control of Transport Networks", and "Common Control Aspects" of the interaction between the ASON control plane, SDN control plane, management plane and transport data plane [172].

2.7 Transport APIs Standardization

Open application programming interfaces (API)s to control and manage transport networks are a major topic of interest for network providers in order to foster programmability to lower CAPEX and OPEX of their multi-layer and multi-vendor transport infrastructure.

The process to define the APIs starts with the definition of a UML information model, from which can be created the data model that will be supported by a protocol like NETCONF or using REST-like protocol running over HTTP to provide the APIs. Thus, we first define information model and the difference with data model:

- **Information model** describes the managed objects (network device or system) at a conceptual level, including the relationships among the objects. The information model is implementation and transport protocol independent. For instance, the Unified Modeling Language (UML) is very common to create an information model.

- **Data model** defines explicitly and precisely the structure, syntax and semantics of the managed objects's information model data. The data model should be complete and consistent. The data model includes protocol-specific rules that explain how to map managed objects onto lower-level protocol constructs.

Programming interfaces towards network elements have been used for a long time by network management systems. The Network management system allows to configure network elements from a centralized entity, similar to the centralized control proposed by SDN. For instance the Simple Network Management Protocol (SNMP) was developed to provide a programmatic interface towards network devices in order to build smart management applications. However,

back in 2002 it was already accepted that SNMP had failed as a network management protocol [173]. The Transaction Language 1 (TL1) and the Common Object Request Broker (CORBA) are two widely used management protocols in telecommunications Networks.

This section summarizes the standardization efforts on transport APIs, which can be classified in two types:

- Northbound Transport APIs (NB-TAPIs): provided by the control plane and standardized by IETF, ONF and OIF (subsections 2.7.1, 2.7.2 and 2.7.3).
- Southbound Transport APIs (SB-TAPIs): provided by the so called white-boxes transport network devices, and standardized by the Open ROADM project (subsection 2.7.4).

In table 2.7 are summarized and compared the main standardization efforts of transport APIs to be deployed at the north bound interface of T-SDN controllers.

Table 2.7: Standardization of Northbound transport APIs (NB-TAPIs)

SDO	Basis	Information Model	Data Model	Interface	Format	References
OIF	Based on ITU-T ASON model [174]	UML	YANG	REST	JSON	[17],[10]
ONF	Based on OIF APIs	UML based on CIM from ITU-T [155] and ONF [154]	YANG	REST and NETCONF	JSON and XML	ONF-TAPI[157] OSSDN [175]
IETF	Based on previous work from IETF	UML based on CIM from ITU-T [155] and ONF [154]	YANG	NETCONF and REST-CONF	XML and JSON	[163, 161, 164, 165, 176, 160]

2.7.1 IETF Transport APIs

From the set of requirements defined in the workshop held by the Internet Architecture Board (IAB) on Network Management [173], the IETF developed the Network Configuration protocol (NETCONF) [149].

NETCONF provides a standard framework and a set of standard Remote Procedure Call (RPC) methods to manipulate the configuration of network devices. In NETCONF, the devices's configuration data, and the protocol data, are encoded with the Extensible Markup Language (XML). NETCONF is primarily transported over the Secure Shell Transport Layer Protocol (SSH).

Nevertheless, NETCONF do not defines the way to express its payload. The YANG data modeling language is standardized in RFC 6020 [4]. YANG provides the means to define the content (both data and operations) carried

via NETCONF [4]. YANG uses XML to represent the contents of the data stores.

Today YANG Data Models are the basis of APIs implementation. YANG is used for REST-based interfaces by encoding the YANG model using JavaScript Object Notation (JSON) text. JSON is a lightweight data-interchange format based on dictionary-like data structure. Draft [156] defines the encoding rules for representing a YANG Data Model as JSON text.

REST is the most used paradigm for definition APIs, it allows to reduce development times and provides multiple debugging tools. Thus, when compared to traditional bit-oriented protocol stacks, REST-like protocols improve development time and offer better debugging capabilities.

The IETF NETCONF Data Modeling Language working group (netmod-WG) recognized the benefits of using a common Information Model (IM) as the foundation to develop purpose and protocol specific interfaces [177].

IETF, ITU-T and ONF adopted the same core IM, ITU-T Recommendation G.7711 [155] and ONF-CIM [154] to boost convergence, interoperability and efficiency of models.

There are multiple works in progress at the IETF to propose T-SDN related YANG data models for:

- Representation and manipulation of Traffic Engineered (TE) topologies [158], interfaces, tunnels and LSPs [159].
- Abstraction and control of TE and virtual networks [160].
- Deployment and operation of transport network open interfaces [161].
- Management of PCEP [162].
- Representation and manipulation of topologies at Layer 1 [163], Layer 0 WSONs [164], and flexi-grid domains [165].

.

YANG has been already adopted by industry-wide open management and control initiatives e.g., OpenDaylight (ODL) and Open Network Operating System (ONOS) (see section 2.8).

2.7.2 OIF Transport API

The OIF was the first to identify that the key to boost interoperability and programmability of transport SDN is the definition of standard north bound APIs. The OIF framework [10] employed the functional elements of the ITU-T ASON model to define the APIs. The OIF APIs are based on REST-like protocol and JSON encoding [10]. The following APIs were defined by the OIF framework for T-SDN:

- Interface to the Call Control or Service request: enables to retrieve connectivity services from the network, such as: creation, deletion, listing and query.

- Interface to Connection control: typically an internal interface used by the service interface to setup the connectivity, however external API can be added for the Connection control.
- Route query or Patch computation interface: allows to request path computation and optimization prior to request establishment of connectivity service. Together with topology interface conform the interface to routing control.
- Network topology interface: enables listing and reading of topology objects directly form the control plane, such as: vertex, edge end, edge, and edge end resource.
- Abstraction control APIs: support of virtualization and abstraction of network resources for specific services. Network abstraction is a representation where some of the topology details are not visible. While network virtualization means a subset of the network resources.
- Notification APIs: retrieves information (or reports) of events, such as alarms, performance monitoring threshold crossing, object creation/deletion, state change, attribute value

A simplified version of the OIF APIs was implemented for the OIF-ONF Global Transport SDN Prototype Demonstration [17] (see section 2.6.2). The service API allowed the same application to be tested across heterogeneous domains. It also allows multiple orchestrators to access the same set of controllers where each orchestrator have access to a subset of resources (virtual slice of the network).

2.7.3 ONF Transport APIs (TAPI) & Common Information Model

The ONF OTWG Transport API (TAPI) project is working in the specification of standard transport APIs. The ONF-TAPI maps to the objects described by the ONF Core Information Model (ONF-CIM) [154]. The ONF-CIM has been developed through collaboration among ITU-T, TeleManagement Forum (TMF), and ONF, and it was published as ITU-T Recommendation G.7711 [155].

The ONF-TAPI project published a functional requirements document for the development of TAPIs [157]. Its main target is to drive the detailed UML information model specifications, from which YANG and JSON data models can be defined to generate Swagger APIs.

Based on the OIF framework [10] the ONF-TAPI proposed a set of interfaces to abstract a common set of transport network control functions:

- Topology Service: retrieve topology, node, link and edge-point details.
- Connectivity Service: retrieve and request point-to-point (P2P), point-to-multipoint (P2MP), multipoint-to-multipoint (MP2MP) connectivity for L0-L1-L2 layers. ONF decided to join the connection control and service request interfaces into a single Connectivity service API.

- Path Computation Service: request for computation and optimization of paths.
- Virtual Network Service: create, update, delete virtual network topologies.
- Notification Service: retrieves information (or reports) of events.

ONF has open sourced the following Open Source SDN (OSSDN) Repositories [175]:

- SNOWMASS project: contains models and code for TAPI, including the TAPI information model in UML, its mapping into YANG and JSON data models/schema, and Swagger REST APIs.
- Englewood project: aims to develop a set of software modules to prototype, test, validate and facilitate the deployment of ONF-TAPIs, in heterogeneous T-SDN environments over open-source controllers (ONOS or ODL), or proprietary platforms (vendor specific SDN controller or legacy NMSes).
- EAGLE project: maintains documents and code of the ONF Information Model Project. Provides open source code to auto-generate YANG model code from a UML code.

2.7.4 OPEN ROADM APIs

OPEN ROADM is a recent initiative by AT&T to define open standards for a disaggregated white-box Reconfigurable Optical Add/Drop Multiplexers (ROADM) [178], so it can be dynamically managed by an SDN control plane. A white-box ROADM provides open APIs towards a transport SDN controller.

The disaggregated ROADM proposed in [166] is conformed by three optical functions: the ROADM switch (optical amplifiers, couplers, and wavelength selective switch), transponders and pluggable optics. Each functional element provides open standard-based APIs towards the transport SDN controller.

The data models of Open ROADM API is written in YANG, and it is planned to have transponders and ROADMs that support YANG models over NETCONF.

- Device model (vendor specific): provides a detailed view of the devices using a generic representation of transponders and ROADMs. Allowing optical equipment vendors to fill the template to describe their ROADMs and transponders. ROADMs can be colorless-directionless or colorless-directionless-contentionless [167].
- Network model (vendor neutral): provides a generic and vendor independent representation of the network. This model is used only by the SDN controller, which needs to map the Device Model into the Network Model [167].
- Service model: service representation at network and Device Model levels.

2.8 Main Open Source SDN Control frameworks

Open source has powered the innovation across many technology fields, and SDN is not the exception. SDN opened the door for Open Source projects in networking, fostering the innovation through experimentation and contribution of a growing SDN community. Such projects are filling the gap of slow standardization process on SDN technologies. There are over 30 SDN controllers on the market today, from open source projects to vendor proprietary platforms. However, in this study we focused on the main open source SDN platforms that can support a control plane capable of managing large service provider networks.

- **OpenDaylight** [31]: designed to serve a broad set of use cases and end user types, but with a main focus on service provider, enterprise, and academic networks. ODL is already a common platform for vendors' solutions, and used in pioneering demonstrations.
- **Open Network Operating System (ONOS)** [32]: a relatively newer player that specifically focuses on carrier networks. ONOS is gaining large momentum among service providers and academy.

ODL and ONOS are both based on Java programming language and the OSGI that provides high modularity and allows loading service specific bundles at runtime. They both support distributed architectures for improvement of scalability and reliability, and provide the largest set of features among the SDN controllers.

2.8.1 OpenDaylight (ODL)

OpenDaylight is an Open Source Software project under the Linux Foundation founded by industry leaders in 2013. The initial aim of ODL is to accelerate SDN development and industry adoption, through the creation of a common industry supported controller platform. Some of the companies contributing to ODL development are: Cisco, Juniper Networks, VMware, Microsoft, and Ericsson. At the moment four releases are available: Hydrogen (Feb, 2014), Helium (Oct, 2014), Lithium (June 2015) and Beryllium that was released in February 2016.

ODL was the first controller that provided a framework to implement control and management services for heterogeneous multi-vendor networks.

OpenDaylight is becoming a de facto standard for SDN controllers with a growing support from the vendor industry that present OpenDaylight-based commercial products (e.g., ADVA, Brocade, Calient, Cisco, Ciena, Corian, Cyan, Ericcsson, HPE, Infinera, NEC, among others that are listed in the Solutions Provider Directory of ODL project [31]).

1. **Control Layer** The main components of ODL are service abstraction layer (SAL), the basic network functions, the enhanced network services,

and the network abstraction (Policy/intent) service functions and pluggable modules. Service Abstraction Layer (SAL) represents a key bundle between service producers and consumers. Modules that provide services have to register their APIs to the SAL registry. Whenever a request from service consumer comes, SAL binds them into “contract". There are two SAL architecture: application driven SAL and module driven SAL.
The model-driven SAL (MD-SAL) framework maintains YANG data structures in a common data store and provides a messaging infrastructure (notifications and RPCs) that facilitates the incorporation of new applications and protocols.
The following basic network functions are preconfigured with the controller: topology processing, OpenFlow statistics manager, OpenFlow switch manager, OpenFlow forwarding rules services, Layer 2 switch an host tracker.
The enhanced network services are platform, protocol and vendor -specific services that provides ODL. ODL supports the largest amount of features among all controllers. Some of the enhanced network services ares: BGP-LS/PCEP, VTN (Virtual Tenant Network) and service function chaining. ODL supports four methods for configuration of policies and intents: Application Layer Traffic Optimization (ALTO), Group Based Policy (GBP) and Network Intent Composition (NIC).

2. **Southbound interface** ODL's MD-SAL and the plug-in model allows to incrementally support multiple south bound interfaces and protocols (vendor-specific and standard). For instance ODL supports OpenFlow, SNMP, NETCONF, OVSDB, BGP, PCEP, LISP and other vendor specific interfaces such as TL1 and CORBA. Commonly, one plugin includes connection, session and state managers, error and packet handler mechanism and set of basic services. Supported protocols communicate to the SAL.
3. **Northbound interface** ODL Controller exposes north bound APIs to the upper layer applications using OSGi framework or bidirectional REST APIs. OpenDaylight APIs can be REST, RESTCONF, NETCONF and AMQP (Advanced Message Queuing Protocol).
On top of ODL APIs, there is a framework for Authentication, Authorization and Accounting (AAA).

2.8.2 Open Network Operating System (ONOS)

ONOS, supported by the ON.Lab [179], is the first Open Source SDN controller to focus on service provider networks [32]. The goals of ONOS as presented in the ONOS whitepaper [180] are:

- A control plane that ensures carrier grade features, i.e., scalable, high performance and five nines availability.
- Enable Web style agility.

- Help service providers migrate their existing networks to *white-boxes*.
- Lower service provider CapEx and OpEx.

Such goals are tackled by the following set of features.

1. **Distributed Core:** The adoption of a distributed core architecture is the key to meet carrier grade requirements, and the main difference with ODL (before the fourth version ODL did not provide clustering capabilities).
   ONOS maintains the centralized logical control of SDN, while running as a service on a cluster of servers, following the approach presented by Koponen et al. in [181].
   A cluster of controllers allows scalability by instantiating control capacity as needed. High availability is provided by a fast failover upon an ONOS server instance failure.
2. **Northbound APIs:** The ONOS north bound APIs hide the complexity of the network and the distributed core. It provides two main abstractions to foster web style agility:
   - The intents framework: allows to request services from the network such as policy statements and connectivity requirements, without dealing with implementation details.
   - The Network view: a consistent view of the element in the network and their related states such as utilization and established connections. A specific API of this abstraction provides a graph representation.
3. **Southbound APIs:** At the ONOS core, network elements are described with generic objects. Using device specific plugins (or south bound providers) ONOS (similar to OpenDaylight), that adapt protocols to the south bound API, can communicate with OpenFlow, NETCONF or other legacy-based protocols like PCEP and TL1. The Southbound API isolates ONOS distributed core from protocols and interface -specific plugins [182].
   ONOS is devoted to the use and creation of commodity hardware and *white-box* devices that can be fully controlled by open and standard south bound APIs.

ONOS vision of the network follows the data center approach of using commodity hardware to bring economy and agility to carrier networks, while avoiding vendor lock-in. In consequence, packet and optical network elements are replaced with low cost *white-box* components, and the central offices are rearchitected as data centers.

While *white-box* packet switches are already standardized and commercialized, the *white-box* optical network elements are in early stages of development.

Among the optical network elements, the ROADM is the key component. ONOS foundation built the first disaggregated *white-box* ROADM using open

source software and commodity hardware from Fujitsu, Ciena, Lumentum, Oplink, and Calient.

The ONOS *white-box* ROADM was disaggregated into three main functional components: transponders, WSS and backplane. The ONOS *white-box* ROADM is based on NETCONF/YANG, and a list of APIs were defined for each component, in order to allow:

- Device discovery.
- Device capabilities detection (ports).
- Device configuration (power, alarms, and transmission values).
- Cross-connection provisioning (configuration of OXC matrix).

As future work, ONOS intent to follow the standardization efforts, that were recently started by the OPEN ROADM project [178] (described in subsection 2.7.4) and the OpenConfig working group [183] (a vendor-neutral, model-driven network management designed by network operators).

The ONOS project has work in progress with optical equipment vendors (Ciena, NEC, Huawei) and service providers (AT&T and SK Telecom). Among the Operator's use cases being developed for ONOS, the following two are of great importance for T-SDN:

1. **Operator Use Case: Packet Optical Convergence**
   As we have exposed in section 2.4.1, SDN allows to gather and manage a converged packet/optical topology. This use case identified the need for providing multi-layer native support in ONOS.
   An overview of their achievement at OFC 2015 [145], included: converged packet/optical network graph abstraction, multi-layer PCE, restoration and protection mechanisms, and development of vendor-specific south bound plugins to enable T-SDN in legacy equipment (providers). An important feature missing in [145] is the discovery of optical layer topology, thus topological information was manually configured.
   As part of this use case, an optical emulator platform called LINC-OE was developed [147]. LINC-OE a software switch that emulates *white-box* ROADMs with extended OpenFlow 1.3 (OpenFlow 1.3+) support.
   In 2015 a demonstration included Ciena and Fujitsu TL1 providers, and Huawei PCEP provider [32], for a real multi-vendor and multi-layer scenario.
2. **Operator Use Case: Central Office Re-architected as a data center (CORD)**
   In order to bring data center economies and cloud agility in carrier networks, while avoiding vendor lock-in, CORD combines NFV, SDN and Cloud using commodity IT and network infrastructure. In a general this use case elaborates in transforming the Point of Presence (POP) and Central Offices (CO)s of operators into mini data centers. Commodity servers are interconnected by a fabric constructed from *white-box* packet switches.

CORD started as an ONOS use case, however it become a full open source project [184], which goal is to: create a reference open source architecture from commodity servers, *white-box* switches, disaggregated access technologies (e.g., virtual Optical Line Terminal, virtual Base Band Unit, virtual Serving Gateway, virtual Router, virtual Packet Gateway), and open source software (e.g., OpenStack, Docker, ONOS, XOS). ONOS/CORD project cover residential, mobile and enterprise domains, each with specific features and configuration.

2.9 Vendor Solutions

By the time of writing this report, transport optical network vendors are at the inflexion point to change from a closed and lock-in prone solutions (*black box* approach) towards a more standard SDN and NFV -based solutions, that should open up control and visibility of their equipment.

On the other hand, packet switched industry is ahead of SDN innovation thanks to a more standard data plane, and there are vendors offering commercial *white-box* products mainly for the data center use case (e.g. Accton, Celestica, and Quanta Computer). *White-box* networking allows to use standard, off-the-shelf switches and routers, and to give full control of such devices to an SDN controller via OpenFlow or other standard SBI. However, the *white-box* approach is much more complicated to accomplish within transport optical network infrastructure due to the challenges presented in section 2.3.3. Nonetheless, advances on SDN and NFV are making possible the concept of disaggregation of functions that are normally integrated into single chassis.

In the transport optical network industry, several vendors already have experimental OpenFlow extensions to support optical domains (mainly for Ethernet and OTN switches). Their main apparent focus is the interoperation between centralized SDN (domain) controller and a distributed GMPLS control plane, or a centralized NMS. The GMPLS and/or NMS serves as the SDN enabler, while on top of it, the SDN domain controller provides NBI APIs to access service request and topology functionalities, mainly down to the OTN layer. Such interoperation is possible through the protocols and techniques presented in section 2.6.3.1. Thus, GMPLS control plane and NMSs are playing an important role for the T-SDN solutions proposed by vendors, following a *black box* approach.

T-SDN must support multi-vendor interoperability in hierarchical architectures. Optical network equipment industry already offers orchestration platforms for network and service orchestration. For instance Ciena Blue Planet [185], Juniper NorthStar [186], Cisco Evolved Programmable Network [12]. A vendor-specific transport network orchestrator could lead to vendor lock-in, which carriers expect to avoid with SDN. A healthy hierarchical architecture should have a vendor-neutral Orchestrator, which can be developed at the application layer, or it can also be based on IETF ABNO [11]. The

network orchestration market is expected to grow fast, the first vendor neutral transport network orchestration is the multi-layer and multi-vendor solution from Sedona [187], which is still tied to a list of vendors.

Tables 2.8 and Table 2.9 gives a comparison of *black box* and *white-box* optical solutions, respectively.

Table 2.8: Vendor Transport SDN solutions - optical *blackbox*

Optical Vendor	Orchestration	Controller or Virtualization engine	SBI	NBI
ADVA	Ensemble Orch. ETSI MANO-compliant. End-to-end VNF and network-service-lifecycle management	RAYcontrol Controller. GMPLS-based. Multi-layer	GMPLS protocols	REST APIs
Alcatel-Lucent	CloudBand NFV Orch.	NSP Controller. NMS + ODL-based. Layer 0-3 support. The NMS configures and manages the optical domain	NETCONF/Yang, PCEP, BGP-LS, OSPF, IS-IS/TE, OpenFlow, SNMP	NETCONF /Yang, REST APIs
Ciena	Blue Planet: Multi-Layer and Multi-vendor Orch., virtualization and management. End-to-end LSO	Multi-layer WAN Controller (MLWC). ODL-based. Layer 0-2 support	TL1 and CORBA (Optical devices). OF, NETCONF, SNMP, PCEP, BGP, OVSDB - Open SBI	REST APIs
Cisco	ESP + Multi-vendor Service Orch.	nLight Controller (IP + Optical). GMPLS-based	PCEP, BGP-LS, RSVP (Optical devices). OF, OpFlex, NETCONF	REST APIs
Coriant	None (Orch. and other business apps. should be build by customers)	Transcend Transport Controller	NETCONF, SNMP, TL1	REST APIs, OpenFlow+
Ericsson	Management and Orchestration	Ericsson SDN Controller. ODL-based. Offers a Transport SDN domain controller Apps.	OF, OVSDB, BGP, NETCONF, PCEP, BGP-LS	REST APIs
Fujitsu	Virtuora Orch.	Virtuora network Controller. ODL-based. Multi-layer and Multi-vendor	NETCONF, TL1 and SNMP	REST APIs
Huawei	NetMatrix Orch.	Smart Network-Transport Controller (SNC-T) ONOS-based (IP + Optical)	PCEP and OSPF for GMPLS domain	REST APIs
Infinera	None	Open Transport Switch (OTS) Abstraction and Virtualization Engine, compatible with third-party Controllers/ Orchestrators.	Vendor specific interfaces	Web 2.0 API
Juniper	Contrail Service Orch.	NorthStart Controller: L1-3 PCE-based controller	PCEP, NETCONF, OSPF-TE, BGP, BGP-LS, ISIS-TE, XMPP	REST APIs

2.10 T-SDN Open Issues

Control, management and orchestration of transport networks is a challenging multi-layer, multi-domain and multi-vendor problem. Such a wide problem, led

Table 2.9: Transport SDN *white-box* solutions

Optical Vendor	**Controller**	**SBI offered by the device**	**Type of device**
Calient	ODL-based Optical Topology Management Controller	OpenFlow V1.3 and V1.4, TL1, SNMP and CORBA	Optical switch [188]
Ciena, Fujitsu & Nokia	Virtuora NC	NETCONF - OpenROADM YANG data models	WSS and Transponders [189]
Lumen	-	OpenFlow V1.4, NETCONF, REST API, SNMP	Optical switch for data center networks [190]
Lumentum	-	TL1, SNMP	Disaggregated *white-boxes* Terminal amplifier, line amplifier, mux/demux, and ROADM-WSS for datacenter and metro edge networks [191]
Polatis	-	OpenFlow, NETCONF, SNMP, TL1, and SCPI	Optical Switch for datacenter networks [192]
Fujitsu	Virtuora NC	NETCONF, SNMP, TL1, and SCPI	1FINITY Metro Data Center Interconnect [193]

to multiple solutions; yet it seems that there may not be a single solution to fit all the scenarios. T-SDN is an open subject with many open issues to be debated within the research community and a very fast innovation pace. This section provides a list of areas in T-SDN architecture that are expected to need significant future work.

2.10.1 Control plane architecture

The architecture of the control plane for heterogeneous multi-domain transport networks is a major concern. Controllers are in continuous evolution to meet the requirements of service providers on availability, scalability and high performance. The Hierarchical control plane architecture (HT-SDN) seems to be the best choice for T-SDN.

The main open source controllers are based on similar internal architectures, but they continue to evolve. In the initial phases of SDN implementation it was important to support multiple south bound interfaces to control green-field and brown-field domains.

For the future, the NBI has become more important. Precisely in order to enable HT-SDN, controllers should become interoperable at the NBI, so that different controllers can speak to the same orchestrator or parent controller. So, some level of agreement must be reached between controller makers about compatible network abstractions, common information models and interoperable APIs exposed at the NBI, which lead us to the next open issues.

2.10.2 Abstractions

To choose the right level of abstraction to understand and fully optimize the transport network resource utilization is the key to future T-SDN success.

A good abstraction layer should have the right balance between: amount of information (complexity) and degree of provided control (flexibility). The optical layer features and impairments must be carefully considered when defining the level of abstraction that will be provided by APIs of T-SDN.

An interesting open issue is to analyze the trade-off between scalability and flexibility of provided abstractions (given by the visibility into the optical domain). In T-SDN the definition of a standard abstraction of the optical layer topology, impairments and complexity remains an open debate. These abstractions can happen at the north bound and south bound interfaces of the T-SDN control plane. Today many vendor solutions provide programmability of the optical data plane, however most of them provide an abstraction view that hides layer 0. Is this the right compromise between flexibility and complexity?

2.10.3 Common Information model

Definition of a common information model (CIM) is the base to build proper technology-agnostic standard abstractions at the north bound and south bound of SDN architecture. ONF is working on the development of a CIM [154].

From a technology-agnostic CIM, a protocol or technology-specific data model can be built. YANG is becoming the de-facto data-modeling language. The APIs are then provided by the data model using a specific format and protocol. The format for north bound APIs is already accepted to be RESTFul, however the data model is still under debate with ongoing research and standardization efforts. Currently, ONF is leading the definition of specifications for T-SDN related CIM [154].

2.10.4 North bound Interface (NBI) APIs

To ease the achievement of multi-vendor, multi-layer and multi-technology T-SDN implementation, the north bound APIs for T-SDN should be standardized.

So that, vendor-agnostic applications and network Orchestration systems, on top of the SDN hierarchical architecture can consume those APIs to deploy full service programmability across heterogeneous domains and layers. The domain controllers can be legacy or OpenFlow-based, and the communication among the hierarchy of controllers can be done using OpenFlow, different flavors of PCE, NETCONF, RESTCONF or just REST APIs over HTTP. However they should provide standard APIs either directly or using adaptation layers towards an application engine that run network Orchestration services.

The process to create the APIs starts with the definition of an UML information model, which can be used as basis to create the data model that will be supported by NETCONF or by a REST-like protocol running over HTTP to provide the APIs. YANG modeling language is the main choice to build technology-specific data models. Abstraction at the north bound provides APIs at the service level, using a graph-like abstraction to provide consistent network-wide view and traffic engineering information (see 2.7.3).

Within the review presented in this paper, several proposals are described, however there is no consensus over this problem. Only recently the standardization bodies and some working groups are looking at standardization of the north bound APIs (see section 2.7)

2.10.5 South Bound Interface (SBI)

The SBI allows the control, management and monitoring of network elements. The standardization of proper SBIs is the foundation of SDN. Today there are multiple SBI protocols: OpenFlow, NETCONF, PCEP, BGP-LS, among many others. One issue is that the optical layer is in continuous evolution, and there are many different vendor-specific implementations.

For protocols like BGP-LS and PCEP there are still multiple IETF drafts to cope with the new advances at the optical layer and in the PCE architecture. Thus, either the protocol implementation is not updated or the controller do not support the new features of the protocol. Standard OpenFlow do not cover yet the full range of properties of optical layers, and there are many non-stable extensions. The technology-specific data models for using NETCONF and REST based protocols are still under development mainly by IETF and ONF.

A very important missing component in today's controllers is generic support for protocols like NETCONF, by providing a data model of a specific network element, for instance implemented using YANG.

In consequence, there is a lot of work to be developed in order to have stable standard SBIs for transport networks.

2.10.6 Scalability and reliability: the distributed controller

The concept of centralization of the control plane is at the basis of the SDN approach. But it should not be forgotten that controllers and orchestrators are software processes running inside a computer platform that have obviously limited resources. Therefore, as the controlled data-plane network gets larger and/or as the number of controlled flows increases, the SDN control plane starts facing the issue of scalability [194]. Moreover, centralization also may imply a reduction of the reliability, whenever single points of failures are generated in the system.

In order to improve scalability in large networks and to enhance reliability, several controllers are not implemented by a software running on a single machine, but by a cluster of machines (e.g. in the Cloud or hosted in the datacenter of the network operator) running a distributed version of the controller [181, 195]. Since the cluster has to behave in all circumstances as a single logical entity, some technique allowing to preserve a constant synchronization between the multiple instances of the controller must be adopted. Usually, such techniques imply the use of some consensus protocol (e.g. the RAFT [196]), supported by an exchange of messages between the remote processes. The logical ports through which the peer instances exchange consensus messages

are called West interfaces (to distinguish them from the NBI and the SBI). Some well-known controllers, such as ONOS [32], were developed to support a distributed architecture since the beginning, some others, such as ODL, are under improvement to achieve or consolidate cluster capability [31].

A more ambitious target, but an interesting opportunity for the future, would be to make different controllers to interoperate exploiting each its East/West interface [197]. Such a solution may be of help in the multi-carrier scenario, where the controllers belong to different network operators, should communicate each other, but on the other hand, disclosing a minimum set of information (e.g. and abstracted topology), controlled according to the inter-carrier policies.

The distributed implementation and its impact on control-plane performance [198, 199] is still a widely open topic, deserving a lot of additional research, especially focusing on the problem of delays (both in instance synchronization and from controllers to devices) and controller placement.

2.10.7 Orchestration

The exact definition of Orchestration and the definition of the precise roles of the Orchestrator is another main open issue. Also regarding this topic, we are in a context largely still uncovered by standardization.

Some optical network equipment vendors already offer network orchestration platforms, such as Ciena Blue Planet [185], Juniper NorthStar [186], Cisco Evolved Programmable Network [12]. However, T-SDN should be based on vendor-neutral Orchestration, which can be developed at the application layer. A few Communication Service Providers are already building their own network Orchestration solutions.

Initial demonstrations of north bound APIs-based orchestration across multi-vendor IP and Optical domains were presented in [107, 187, 200]. In [187] vendors created an adaptation layer to populate a common API defined by SEDONA systems. In both demonstrations the orchestrator do not have control over the physical layer. In early 2016, the Linux Foundation formed the OPEN-Orchestrator Project (OPEN-O) to develop the first open source software framework and orchestrator for agile operations of SDN and NFV [128]. ONOS is also developing an orchestration platform for the CORD project to provide everything as a service (XaaS) exploiting SDN, micro-services and disaggregation using open source software and commodity hardware [184].

This context of uncertainty leave a degree of freedom on where to implement the control plane intelligence whether in the controller or in the orchestrator. The decision between controller or orchestrator is not irrelevant, especially as it may greatly influence inter-domain interoperability in the HT-SDN. The more functions are implemented in the controller, the more the entire control plane becomes locked into a specific controller, and the orchestrator will probably have hard time to make it interoperate with others. On the other hand, the simpler are the controllers, the more complex become the orchestrator, with

the risk of being severely limited in scalability. This trade-off is somehow similar to what happens with controllers and data equipment, but at a higher abstraction layer. Up to date, the impression is that even very evolved and complex controllers such as ONOS and OpenDaylight have not reached yet a very stable stage of development, and thus a lot has to be implemented in the orchestrator to customize it to the operators' needs.

2.10.8 Algorithms

The graph-based view of the network gathered at the control plane or at the orchestrator, allows the deployment of applications that run optimization algorithms for multi-domain and multi-layer networks. Such algorithms, previously hidden inside vendor-specific solutions, can now be proposed by third-party entities. This will foster the formation of a scientific ecosystem, which surely benefits from academia especially in developing and demonstrating operational research algorithms for: resource allocation, restoration, resiliency, disaster recovery, virtual network embedding, using a wide variety of architectures, for instance HT-SDN with and without NFV.

In this context there are elements of change compared to the past that will generate innovation also on the scientific ad mathematical side. In fact, a lot of work has been done in the past to develop algorithms tailored to the distributed control planes, such as GMPLS. Now, this previous work has to be retuned to the SDN centralized conception, however taking into account that centralization is at a logical level, while it has to be backed by distribution of process in a cluster for scalability and reliability, as explained in Sec. F. Algorithmically, it is surely an interesting challenge.

The *white-box* approach followed by the OPEN ROADM MSA and other players such as ONOS/CORD, moves L0 complexity management from the devices to the controller so that, even in the lower control layers, some development of standard algorithm to cope with the analog nature of the data plane is necessary.

2.10.9 T-SDN, NFV and security

Service providers are implementing and testing NFV before T-SDN. However, the virtualization of network functions in a T-SDN enabled network is a promising area in very early stage of development. T-SDN and NFV can be used to foster flexibility, agility and resiliency. For instance, a virtual controller can be scaled up/down or even relocated based on network conditions, and upon failures and disasters. The CORD project, is a powerful general-purpose platform to leverage T-SDN and NFV innovation [184].

Security is a big issue in T-SDN. Multiple tenants can share the same infrastructure at different levels. It is still an open area to analyze how SDN principles can be applied to improve security of the tenants. For instance: network slices must provide isolation degrees, based on service level agreements,

and authentication services to the clients. SDN architecture, introduce both threads and solution capabilities on security due to the centralized control plane [14, 201]. Transport networks, manage high capacity and cover long distances, allowing to large-scale attacks. However, security for T-SDN scenarios has not been studied.

T-SDN, NFV and security are the basis of software defined WAN (SD-WAN), a technology that creates programmable overlay networks among enterprise customer premise entities (CPE). Ahead of service providers adoption of T-SDN and NFV, SD-WAN brings agility and programmability for enterprise networks, adopting T-SDN principles on overlay networks. The adoption of T-SDN and NFV in telecommunication service provider networks will play a very important role towards the necessary evolution of their services.

2.10.10 Migration Path towards T-SDN

In principle, the SDN-architecture allows the control and the data planes to evolve separately. However, as we have shown in the previous sections, the two main obstacles that may jeopardize the fast deployment of T-SDN in case of the transport network may be summarized as follows:

- The adoption of SDN requires that data-plane devices are SDN-enabled: this may require in some occasion an investment by the operators to update their equipment that may be hard to sustain
- The heterogeneity and the complexity of equipment, and in particular of the photonic switches, makes it difficult to develop a once-for-all controller and obliges to implement expensive ad-hoc developments on the SBI

These two issues are accompanied to the lack of standardization on the NBI that we have already discussed in the previous sections.

The migration from legacy transport networks to T-SDN is still an open issues that today face telecommunication providers [202]. Hybrid T-SDN deployment is expected to be the most promising solution, as it allows to exploit legacy control-plane solution, such as GMPLS/ASON and PCE architectures, without replacing equipment. Segment routing represents another interesting technology ensuring a migration step towards TSDN. It allows implementing SDN concepts into MPLS-based networks, with multi-layer and multi-domain capabilities (see section 6).

On the side of SBI and NBI standardization, the new concept that is now gaining much attention and popularity is the *white-box*. Recently disaggregation of optical devices is gaining attention for leveraging modular *white-box* building blocks (see table 2.9). For instance, vendors like Lumentum already have commercial disaggregated *white-box* building blocks, including: terminal amplifier, line amplifier, mux/demux, and ROADM-WSS for datacenter and metro edge networks [191]. As consequence, standardization of disaggregated *white-box* ROADMs has just begun with the OPEN ROADM MSA activities

[191]. Service providers are willing to deploy *white-box* devices into their optical transport networks, so the full capabilities of SDN and NFV can be exploited.

2.11 Concluding Remarks

2.11.1 Summary

This chapter provides a comprehensive survey on Transport SDN. To recapitulate, the main points of the evolutionary story we have presented can be summarized as follows.

To offer transport connectivity, a transport-network provider must control multi-domain and multi-vendor network in some cases composed by diverse optical technologies. This complexity seriously challenged the model of SDN as it was conceived for purely-packet and datacenter networks. SDN had to be reviewed before extending it to transport networks.

Since most of the challenges to SDN derived from the optical technology, enabling SDN into optical networks (SDON) was the first step towards T-SDN. The process of effectively matching SDN and optical networks is still ongoing.

SDN principle of separating control plane from the forwarding devices relies on the creation of standard interfaces between these two planes, *i.e.*, standard south bound interfaces. However, realizing standard SBIs means to uniformly abstract the heterogeneity of implementations. Transport-network equipment manufacturers (especially, optical-equipment vendors) have added value to their solutions by introducing innovative features and device capacities that differentiate their products from other vendors: that partially clashes with the concept of uniform abstraction.

A solution to this contradiction is the hierarchical control plane (HT-SDN) paradigm. An operator can manage multiple domains, where domain-specific controllers provide abstracted views towards higher order controllers or network orchestrator, using north bound interfaces. HT-SDN is well supported by standardization bodies (e.g., ONF and OIF) and vendors, and has been the architectural choice for T-SDN research efforts. The separation of control and data plane allows the orchestration of end-to-end services across domains, using abstracted views provided by the north bound APIs of the control plane. HT-SDN is also a promising solution to the co-existence of SDN with other legacy but widespread control-plane implementations, such as GMPLS. So, it is also the key to accelerate T-SDN deployment.

Transport network orchestration must be ideally vendor agnostic, avoiding vendor lock-in supported by standard NBIs. Thus, HT-SDN moves the issue of standardization from SBI to NBI: standard north bound APIs (also called transport API or TAPI) are needed to leverage multi-domain and multi-vendor interoperability and independence of orchestrators from controllers. The TAPI must provide the right compromise between complexity and flexibility in the transport network. This approach can generate the ecosystem of network-software developers, perhaps led by open-source communities and projects,

independent from vendors and network operators, competing to offer orchestration and application at lower prices than traditional vendors. At that point, the promise of T-SDN will honor the promise of being a cost-saving technology, as it happened in the datacenter world. The last evolutionary step is the integration of SDN and NFV to foster the deployment of control plane and virtual data-plane network functionalities, adding all features (e.g. security) to provide services to final customers (business, residential, mobile, etc.), as we have explained in Sections 2.4.3.3 and 2.10.

2.11.2 Final Comments

At the conclusion of this long journey through the history of T-SDN, we can conclude by underlining a remarkable aspect of the adventure of this new technology. That is the extreme dynamism of the SDN concept that is able to quickly reshape attitudes of scientific and industrial world that previously seemed to be consolidated from ages.

Let us mention for instance the attitude towards standardization. In pre-SDN age, the concept of openness of a system was strictly related to coding everything into an official standard, possibly also with legal implications. With SDN, everybody started drifting from this attitude, to exploit quick-and-dirty solutions backed by a release of some widespread software (such as is, for instance, OpenFlow). But with T-SDN, early deployments soon revealed that the lack of standard was not appropriate to control multiple domains and so T-SDN is getting back to standardization. Also ONF is now supporting standard development by its transport working group, oriented to develop a common information model and standard interfaces (e.g. the standard TAPI effort, jointly with OIF).

Similar comments can be made about central vs. distributed control: we started from the all-distributed paradigm of the Internet protocols, to move to the absolute centralization of SDN; but with T-SDN (and SDON) again there is a drift back to distribution (at least, per-domain), as testified by the HT-SDN architecture, and by coexistence with some distributed control plane such as MPLS and GMPLS/ASON through protocol extensions for PCEP, BGP-LS and Segment Routing.

This swinging between attempts to escape ossification and roadmap corrections after reality checks indicate that SDN can indeed prove to be a disruptive technology also for transport networks.

T-SDN is a reality: it cannot be clearer that there is a huge demand for T-SDN. It gained big momentum in the last years, with big efforts from academia, industry, standardization and open source communities. However, T-SDN is just in an initial stage, and there are many open issues to be solved. There is a long path before stable standards will rule the implementation of T-SDN. Therefore, it is really fascinating and exciting for researchers to be part of this evolution, but - more important - economically vital for transport-network operators.

Energy efficient dynamic optical routing for mobile metro-core networks under tidal traffic 3

In general, humans follow highly predictable daily movements. We commute from residential to working and/or educational areas in a daily basis, and we have a selection of commercial and recreational areas for the nights and weekends. We also use the mobile phone at regular hours, for example when commuting, during lunch break and at night. Such regular behavior creates predictable spatio-temporal fluctuations of traffic patterns, which in analogy to the periodical rise and fall of the sea levels, its known as tidal traffic phenomenon [24]. In this Chapter, we exploit such predictability, and the centralized control plane and programmability of SDN we propose off-line optimization and on-line matheuristic to reduce the energy consumption in the optical layer of metro-core networks. We propose a suite of on-line matheuristics to reduce energy consumption of optical layer in mobile metro-core networks, while providing 1+1 protection of the aggregated traffic. The proposed matheuristic dynamically optimizes resource allocation by effectively adapting to predictable aggregated tidal traffic variations. The optimality of on-line decisions is achieved by generating optimal weighted graphs from the interaction with an off-line optimization phase. In the off-line phase either a wavelength path (WP) or a virtual-WP (VWP) problem is solved using the regular tidal traffic patterns. Our results display energy savings of more than 20% by introducing load adaptive network operation, while the matheuristic closely follows the optimal results. We also introduce a heuristic method to reduce service disruption due to routing changes, while preserving energy saving capability. The heuristic allows to reduce service disruption in the range of 38 to 80% with a small penalty on energy saving of less than 5%.

3.1 Introduction

Today, the volume of mobile data traffic (cellular) is by far smaller in comparison with its fixed counterpart, but it is growing two times faster. With an expected 10-fold growth from 2016 to 2021 [21], by 2020 mobile data traffic will represent 15% of global IP traffic [203].

The popularity of smart-phones, the evolution of 4G and the advent of 5G systems, HD-resolutions mobile-terminal screens, mobile cloud services and the Internet of things are major contributions to the expected mobile data traffic. Since the beginning of 2016 commercial 4G devices support downlink data speeds of 1 Gbps [21], and among the requirements of 5G it is expected a 1000-fold growth of mobile access bandwidth per unit area from 2015 to 2020 [22].

Mobile data traffic refers to data traffic over cellular networks such as 2G, 3G and 4G radio systems. It is important to notice that today, mobile data traffic does not include the traffic from "wireless" systems such as Wi-Fi, which provides a "wireless" access to a fixed Internet connectivity based, for instance, on xDSL (Digital Subscriber Line technologies), coaxial cable, fiber or even optical wireless access. However, 5G will be the first mobile technology that takes into account the convergence with fixed wireless (fixed and mobile convergence).

Tidal traffic may create hot spots in the network that move in the spatio-temporal space, following a regular pattern given by the human commutation from residential areas to working areas (academic, business, industrial, medical, governmental among others). Special events may change the tidal traffic, as for instance maintenance, disasters, and social events.

Daily movements of large populations of citizens in urban areas are highly predictable [23]. Moreover, it has been shown that the daily variation of the number of users in a specific area of the city is very periodic, so that the regular traffic daily pattern can be even used to identify the social function of the urban zone [204, 205].

Detection of tidal trends in the traffic load of the mobile-network cells is a valuable knowledge that allows to improve the network resources management. Per cell traffic prediction was used to activate and deactivate or wake-up and sleep the network's cells to increase energy efficiency [206] [207].

In this work we evaluate how human mobility patterns observed at the cells of a radio access network can influence the tidal variations of mobile metro-core network aggregated traffic load (see Fig. 3.1). The recognition of regular components in aggregated traffic allowed us to propose two off-line optimization methods for dynamic routing and resource allocation of mobile metro-core networks [208], that reduce the energy consumption of the optical layer. We have also proposed on-line matheuristics to cope with unpredictable traffic fluctuations [209]. Our heuristics reuse the off-line optimization results over regular traffic patterns to improve optimality.

The Chapter is organized as follows: section 3.2 provides a brief overview of related works and main contribution. In section 3.3 we define the MCN, a procedure to synthesize MCN topology, the mobile-traffic aggregation process which occurs from the cell base-stations to the core-network access nodes and we show the effects of human-mobility patterns to create the aggregated tidal traffic. The proposed off-line and on-line methodologies for energy efficient resource allocation based on traffic grooming and optical bypass are described in sections 3.4 and 3.5, respectively. The power consumption models and the numerical results are reported in section 3.6. Finally conclusions and future works are presented in section 3.7.

3.2 Related works

The mobile carrier network (MCN) traffic can be used to analyze the movements of human beings in a specific area. In [23], a human mobility analysis from a MCN shows that there is a high predictability (from 80% to 93%) on the user mobility due to the inherent regularity of human behavior.

The human mobility is related to the traffic demand variation of the MCN. A study from Google [210], showed that human mobility is strongly related with time-dependent traffic fluctuations at radio access network, neglecting its spatial variations.

The energy efficiency limits of networks that can adapt to traffic load variations in time was analyzed in [20]. Most of the works on energy efficiency takes into account only the temporal fluctuation of the traffic demand. Given that base stations are the most power-hungry devices in MCN architectures, the energy efficiency efforts in MCN focused mainly on the radio access segment [206, 207].

The combined effect of temporal and spatial traffic was first proposed for energy efficient operation of access networks. Starting from the radio access networks with a small cluster of cell sites [24]. [211] proposed a energy efficient management for passive optical network. Recently, tidal traffic effect was considered in [212][213] to propose energy efficiency in metropolitan networks. A limitation of this works, is that they assumed mainly two basic tidal traffic patterns: residential and business. However, the social composition of metropolitan areas is more complex than just residential and business, and multiple social functions or services can coexist in the same location. In this work, we start from the human mobility patterns to build the traffic load patterns, allowing to synthesize more complex tidal patterns on the city.

Currently the contribution of mobile data traffic in backbone networks is very small when compared to the load generated by fixed connections. Therefore, energy efficiency approaches did not consider the traffic generated by mobile networks [214, 215, 216, 212, 213]. However, it is expected that in the year 2020 mobile data traffic will be increased by up to thousand times its current volume, reaching 100 Gbps per square kilometer and 500 Gbyte/user/month

[22] [214]. Recently, [217] analyzed the consumption of the back-haul segment of heterogeneous mobile networks, that is said to become one of the bottlenecks of future mobile networks.

3.2.1 Contribution

In this Chapter we propose a suit of methodologies that exploit the presence of regular aggregated-traffic patterns in the MCN and employ short-term traffic predictions to improve the use of dynamic optical resource allocation. However, traffic pattern recognition, and traffic demand prediction techniques are not the focus of this work. Our main focus is the optimization of dynamic resource allocation techniques to minimize the energy consumption of optical metro-core networks.

The contribution of this study is to propose off-line [208] and on-line [209] optimization methods that exploit regular components of aggregated mobile tidal traffic to improve the energy efficiency of the mobile metro-core network while providing 1+1 protection.

In [208] we proposed two off-line optimization methods that target predictable aggregated mobile traffic in urban areas. In [209] on-line optimization matheuristic (Algorithm 2) for dynamic optical routing in mobile metro-core networks, able to cope with predictable tidal traffic patterns with unpredictable fluctuations. The matheuristic is based on two phases: 1) *Off-line phase*: exploits tidal traffic phenomenon to solve an optimization problem based on predictable traffic patterns. 2) *On-line phase*: reduces the optimality gap of dynamic resource allocation by solving a simple link-disjoint path-pairs algorithm over an optimal weighted graph that is calculated with the off-line phase results. We have also proposed a scheduling heuristic that provides a good trade-off between reduction of routing changes (to avoid disruption) and resource allocation efficiency (to increase energy savings), applicable to the on-line optimization matheuristic.

3.3 Human mobility patterns and tidal traffic demand generation in MCN

We have used a MCN real dataset that contains anonymized detailed call records of 4869 cells from a mid-sized city of China, over a two-month time period, and real cell site geographical locations. This section starts defining the general MCN architecture. Then, it describes a procedure to synthesize the MCN network architecture. Finally, it provides characterization of human mobility patterns and traffic generation methods that we have used.

3.3.1 The mobile carrier network (MCN)

In this work we based our assumptions on LTE current deployment, but the same architecture remains valid also to support a future evolution to 5G.

As depicted in Fig. 3.1, the MCN is commonly modeled with a three-level hierarchical architecture, composed by the *radio access*, the *back-haul* and the *backbone* networks [218] [219]. The *base stations* (BS)s deployed on the field provide wireless access. Each BS (comprising a set of cell antennas and an eNodeB) is connected by a back-haul network segment to an *aggregation node* (AN). The ANs are the edge elements interfacing the back-haul to the backbone network. The backbone network is the infrastructure connecting the ANs to the *serving gateway* (SGW). We assumed that the metro backbone is divided into an aggregation and a core segment, in order to gradually groom traffic of the metro area from the edges towards the SGW. The *aggregation network* is composed by metro optical rings, each one connecting a subset of neighbor ANs. On every ring, two ring nodes, called *interfacing nodes* (IN)s, are used to interface the aggregation to the core segment of the backbone network. Each IN is connected to a *core nod*e (CN) of the core network (Fig. 3.1). The *mobile metro-core network* is the mesh-topology fiber infrastructure interconnecting the core nodes and the SGW. The assumption of mesh topology of the core is consistent with the current evolutionary trend leading from ring to mesh in the metro areas, given the abundance of fiber links in large cities. We suppose that each node of the core network is an optical cross connect (OXC). The combination of aggregation rings and mesh core, together with the dual-homed interconnection of each ring to the core, enables full resilience of the physical infrastructure of the entire mobile metro backbone against (at least single) failures.

The SGW is connected to a *packet gateway* (PGW) which provides connectivity towards data center facilities and Internet Exchange Points (IXP)s. It is important to notice that all the mobile data traffic must pass through the metro SGW: therefore the part of network which extends from the SGW to the PWG and beyond has not been included in this study.

Given that the connections in the MCN transports large volumes of traffic to/from aggregation rings, and should meet service level agreements to offer carrier grade services, we assumed that these connections need to be provisioned with 1+1 protection scheme [64]. The combination of aggregation rings and mesh core, together with the dual-homed interconnection of each ring to the core, enables full resilience of the physical infrastructure of the entire mobile metro backbone against (at least single) failures. The methodologies proposed in the following sections introduce 1+1 protection with a pair of edge disjoint paths obtained with Algorithm 1, which establishes the active and backup path on different INs of each ring. Offering 1+1 protection is still simple, because it can be seen as the establishment of two link-disjoint dedicated connections for each request instead of a single path. Moreover, we avoided the shared path protection scheme to keep the problem simpler [220].

The MCN topology was synthesized from the real geographical location of 4869 BSs of an anonymous Chinese city. *RAN*: starting from the BSs, we used a clustering algorithm that creates groups of 10 to 12 BSs by minimizing the distance between BSs and the AN of the cluster.

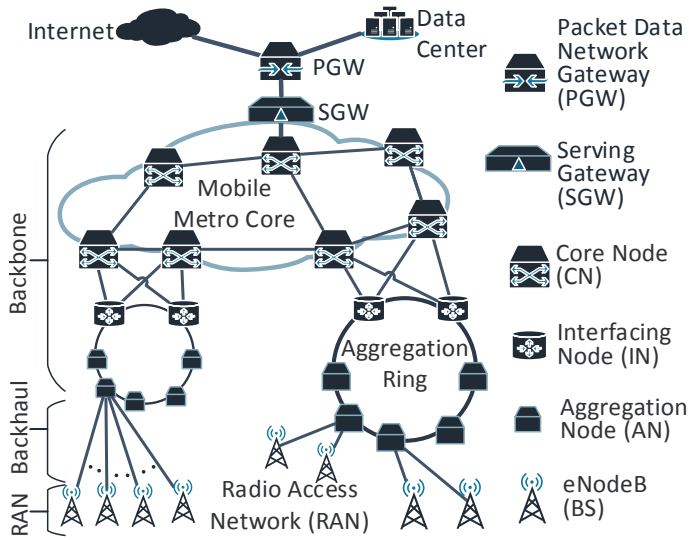

Figure 3.1: Reference mobile carrier network (MCN) architecture.

Aggregation rings: from the set of ANs, a second level of clustering was performed to create 23 aggregation rings of 20 ANs each.

Metro-core: from each ring two nodes where marked as the INs. The metro-core is composed by 46 CNs and one SGW, connected by multi-fiber links (with 80 wavelengths per fiber) in a maximal planar graph with degree 6. Finally, we set the number of fibers of each link of the metro-core network to minimize CapEx, assuming every ring is generating traffic at the daily peak. This dimensioning was done by solving two multi-commodity problems using VWP and WP (for simplifying the presentation, we do not describe these formulations, while VWP and VP are introduced in section 3.4). The resulting network for WP model is shown in Fig. 3.2 [1].

3.3.2 Human Mobility Patterns

A human mobility pattern is defined by the time people reach (leave) a region and the place people come from (leave for). As demonstrated in [23], humans in urban areas display regular patterns of behavior with up to 93% of predictability, which can be converted into mobility patterns by data-mining algorithms. In this study we are interested into the spatio-temporal profile of number of mobile users $N_{c,t}$ in each cell c of the mobile metro network at each hour t of the day. $N_{c,t}$ directly generates spatio-temporal traffic load fluctuations at the cells of the MCN. To this end, we have used a dataset from a mobile network that

[1] Aggregation rings are shown just for an illustrative purpose but they are not actually part of the metro-core network

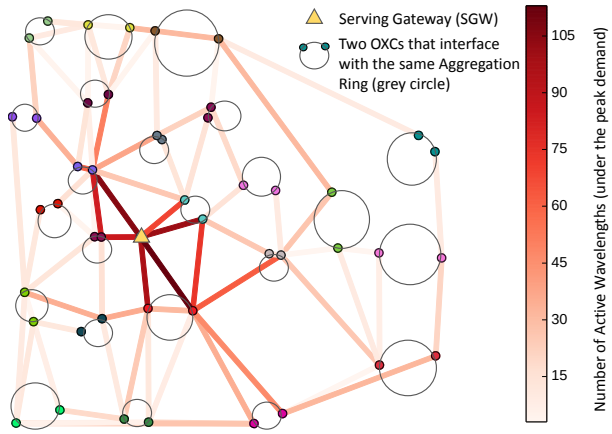

FIGURE 3.2: Synthetic topology of the Chinese city based on real location data and a conservative dimensioning on the peak.

contains anonymized detailed call records of 4869 cells from a mid-sized city of China, that includes information on weekdays over a two month time period.

Fig. 3.3 shows the average daily number of mobile users $N_{c,t}$ in some sample cells of the city. A substantial correlation has been observed [204][205] between the profiles of these curves $N_{c,t}$ and the social functions (i.e., residential area, educational zones and commercial districts) which are prevalent in the area covered by the cell. For example, cell 1515 can be located at a residential area where the $N_{1515,t}$ is high at night time and low during day time (around 08:00 to 19:00). The rest of the cells display an opposite behavior with lower number of users during night and peak values in day time. We can also notice that during morning time there is a clear movement of people from residential areas (cell 1515) towards other social function areas (cells 1934, 709 and 4406). After 16:00 the mobility pattern shifts as the inhabitants of the city go back to their residential areas: thus while $N_{1934,t>16}$, $N_{709,t>16}$ and $N_{4406,t>16}$ decreases, $N_{1515,t>16}$ increases.

3.3.3 Per Cell traffic generation

To test the impact of human mobility patterns in the aggregated mobile metro-core traffic, and evaluate the effectiveness of a load-adaptive network operation, a time and cell (location) dependent traffic matrix has to be generated. We know the daily profile of the number of users in each cell $N_{c,t}$, but we need to convert this data into the total amount of up-(down-)stream traffic in the cell. Several studies show that the personal average usage of Internet services is not constant during the day, but follows an hourly profile ρ_t. We took as a reference the usage profile presented in [221], but many other papers show curves that are very similar. We assume the “desired" traffic, i.e. the unconstrained traffic

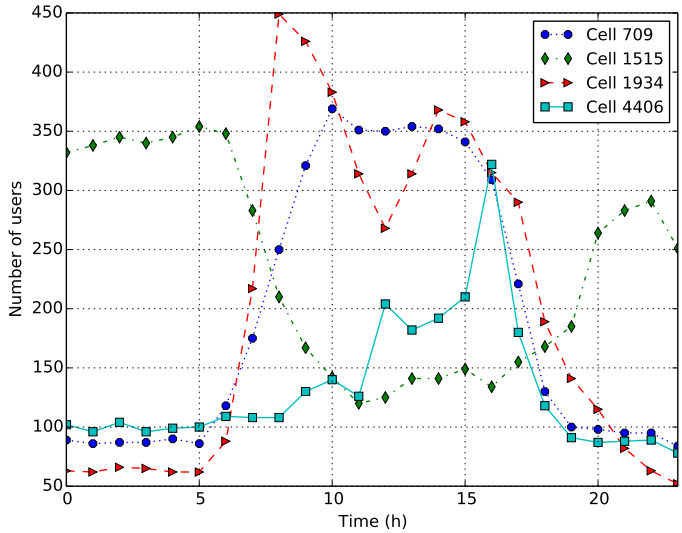

Figure 3.3: Average number of users $N_{c,t}$ within 4 cells of the Chinese city.

demand, C_t^d of the average mobile user of our city at time t, is given by (3.1):

$$C_t^d = \rho_t \frac{\varphi}{30} \frac{8}{3600} \quad [Mbit/s] \tag{3.1}$$

In (3.1), ρ_t represents the normalized traffic intensity per user at time t, and φ is the average traffic volume per active user per month. ρ_t in Equation (3.2) was taken from [221], which analyzed real traffic data of an entire user population.

In order to model the radio access network traffic, one must consider that mobile traffic is elastic, i.e. the load offered by a user connected to a cell c is constrained by the current available capacity in c. The traffic load $L_{c,t}$ generated by $N_{c,t}$ users in cell c at time t can be modeled by equation (3.2)

$$L_{c,t} = [N_{c,t} \cdot \min\{C_t^d, C_{c,t}^a\}] \quad [Mbit/s] \tag{3.2}$$

where C_t^d is given by (3.1) and the available capacity per users $C_{c,t}^a$ is calculated by (3.3)

$$C_{c,t}^a = C/N_{c,t} \quad [Mbit/s] \tag{3.3}$$

where C is the maximum mobile cell throughput at packet layer, i.e. the maximum packet traffic volume which can be fed from a cell to the back-haul link via the S1 interface [2].

[2]In this Chapter we always assume a constant ratio between uplink and downlink traffic of 1/4 (see Sec. 3.3.4). Also, we assumed the spectral efficiency of the radio access to be maximum.

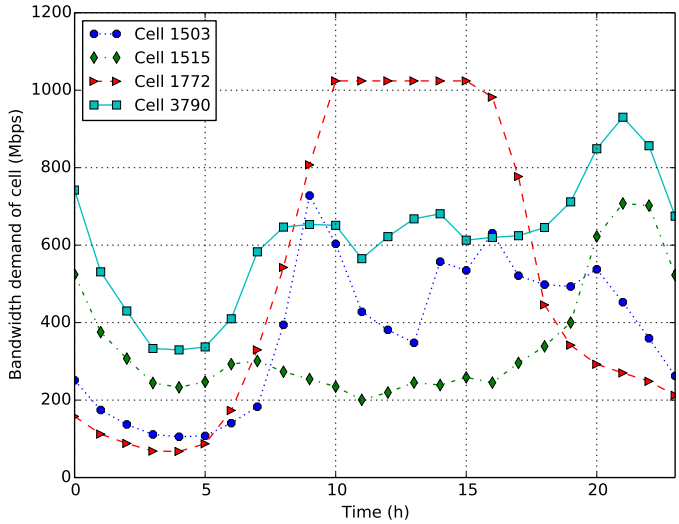

FIGURE 3.4: Generated traffic load $L_{c,t}$ samples within 4 cells of the Chinese city.

According to a 5G European project METIS [22] and Ericsson forecast for 2020 [21], we have set the cell throughput at $C = 1024$ Mbits/s, and the average traffic volume of an active user at $\varphi = 500$ Gbyte/user/month. A sample of the generated traffic load $L_{c,t}$ using (3.2) is shown in Fig. 3.4.

The variation of traffic load $L_{c,t}$ in the cells is given by the values of $N_{c,t}$ and ρ_t as modeled by equation (3.2). For example, cell 1515 (our representative for residential area) experiences highest traffic load during night time, though in Fig. 3.3 the number of users is also high between 00:00 and 06:00. The former behavior of $L_{1515,0\leq t\leq 6}$ is expected as the normalized traffic intensity $\rho_{2\leq t\leq 6}$ is at the lowest value from 02:00 to 06:00, when all the cells experience a load decay. A cell that reaches its maximum capacity C is said to be saturated. Cell 1772 saturates throughout the course of the working period, suggesting a busy educational, commercial or industrial area.

Fig. 3.4 depicts the shift of traffic from some cells to others while the distribution of users $N_{c,t}$ in the city changes over time, e.g., while $L_{1772,t>16}$ starts to decrease, $L_{1515,t>16}$ and $L_{3790,t>16}$ start to increase.

3.3.4 Aggregated Tidal Traffic Generation

The dependence of the traffic on the number of users at the radio access network level is quite obvious. However, the impact of such variations at the aggregation level of an entire ring (with more than 200 cells) is not so obvious. We explore it in this section.

In the backbone network architecture defined in Sec. 3.3, each metro ring

grooms traffic from a set of ANs. Each AN aggregates a cluster of eNodeBs selected on the basis of their geographical proximity in the city area (the locations of the cells are known and a clustering algorithms selects the cells to be connected to the same AN). Also ANs connected to the same ring are selected on the basis of their proximity. Thus, each ring $j \in J$ aggregates traffic of a specific set of neighbor cells F_j. Based on the filtering performed by F_j selection and on the traffic load per cell $L_{c,t}$, the total bandwidth volume $H_{j,t}$ of each aggregation ring j at time t is calculated by (3.4)

$$H_{j,t} = (1 + \Theta) \sum_{c \in J} L_{c,t} \quad [Mbit/s] \tag{3.4}$$

where Θ is the overhead introduced by transport protocols. For LTE, Θ takes values between 10% and 25% [222]. $H_{j,t}$ is composed by up-link (UL) $H^{UL}_{j,t}$ and down-link (DL) $H^{DL}_{j,t}$ traffic. In general, the UL to DL ratio ($H^{UL}_{j,t}/H^{DL}_{j,t}$) observed in a mobile network is 1/4 [21].

Equation (3.4) generates the predictable traffic component on a mobile metro core network based on human mobility patterns. Fig. 3.5 depicts the down-link traffic of three rings $H^{DL}_{j,t}$ (in total, the backbone network includes 23 aggregation rings). The aggregated traffic of each of the three rings decreases from $t =$ 00:00 to $t =$ 04:00, and increases from $t =$ 04:00 to $t =$ 10:00. More interesting variations are observed during $10 < t < 20$. In this period, we could find quite clear evidence that traffic is dependent on user mobility even at the aggregated level.

By applying the methods proposed in [205] we can obtain the composition of rings in terms of social functions. Ring 7 covers mainly scenic/historic regions that people use to visit during the day: observing the curve in Fig. 3.5, $H^{DL}_{7,t}$ starts to decrease at $t = 16$, when people move out of those regions, in opposite to the average tendency that starts to decrease at $t = 20$. Ring 22 has similar behavior as Ring 3: both span on commercial/entertainment districts, but Ring 22 has high proportions of health and educational districts, while Ring 3 is mainly composed of hotels/motels districts. Therefore, Ring 3 has less traffic in the morning, when usually guests leave the hotels, while it presents a steady increase $H^{DL}_{3,15>t>20}$ to reach its maximum at $t = 20$, as guests move back in. Ring 22 displays a flat-like trend in the working hours $H^{DL}_{22,10<t<20}$.

3.4 Offline Mobility and Energy aware Optimization Procedure

The role of the core network is to connect CNs to the SGW (see Fig. 3.1). Therefore, the mobile metro-core network has to satisfy all up-link (UL) and down-link (DL) demands between aggregation rings and SGW. We assumed all these connections are provisioned with 1+1 protection scheme.

The common practice in MCNs is to perform a static resource allocation to meet the peak-hour demand. This method leads to poor energy efficiency,

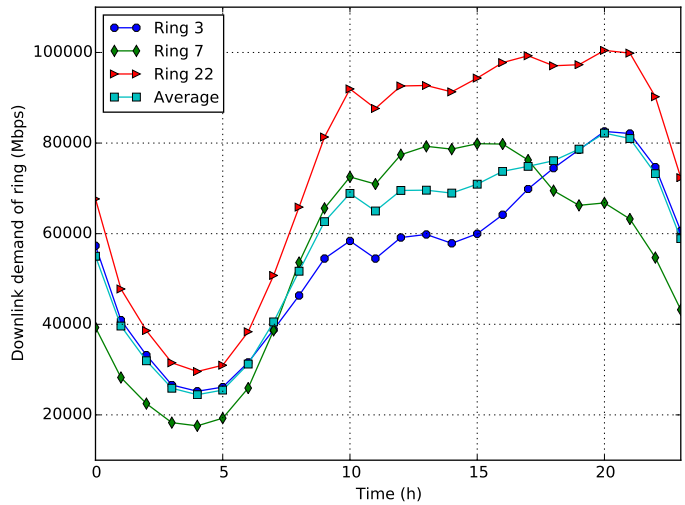

FIGURE 3.5: Predictable aggregated tidal traffic demands (down-link direction) $H_{j,t}^{DL}$.

as outside the peak hour resources will be over-provisioned. As reported in section 3.3, the aggregated traffic demand of a MCN can be predicted from the observation of citizen mobility patterns. By exploiting tidal traffic predictability, an operator can use optimization procedures to dynamically adapt the used resources to the actual hourly need, thus reducing energy consumption.

We proposed two mixed integer linear programming (MILP) models to minimize the energy consumption of the mobile metro core network (as depicted in Fig. 3.1) by activating and deactivating resources every hour based on the predicted component of the traffic demand $H_{j,t}^{UL}$ and $H_{j,t}^{DL}$.

The optical metro-core network can be represented by a bi-directed graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ are the sets of nodes (OXCs) and (directed) fiber links, respectively. Each demand $d \in \mathcal{D}$ is described by the source and destination pair, and the throughput of such demand $h_{d,t}$ (in Gbit/s) which varies in time t. Each demand $d \in \mathcal{D}$ requests $r_d = \lceil h_d/L \rceil$ wavelength connections (where L is the line rate) which can be split and provisioned with 1+1 protection (two link-disjoint paths per connection $r \in r_d$). A demand $d \in \mathcal{D}$ can be Up-link (UL) from an aggregation ring $j \in \mathcal{J}$ towards the SWG or Down-link (DL) from the SWG to j. As depicted in Fig. 3.1 each aggregation ring j has two interfacing nodes $v \in \mathcal{V}$.

The off-line planning problem consists in finding the set of paths that satisfies the spatio-temporal -dependent demand matrix of a specific time period t using 1+1 protection, with the objective of minimizing the energy consumption of the optical layer of the mobile metro-core network.

In the following subsections we present two path formulations for optimal

Algorithm 1 k pairs of link disjoint path: k-PLDP(src, dst, k)

Given the original graph $\mathcal{G}$, create $\mathcal{G}'$ by introducing a dummy node j $\mathcal{V}' = \mathcal{V} \cup j$, and two directed dummy links $\mathcal{E}' = \mathcal{E} \cup (e^{d1}, e^{d2})$ that connects with the two interfacing nodes v of the aggregation ring j. For DL $src = sgw$, $dst = j$, and e^{d1},e^{d2} are incoming links of j. For UL $src = j$, $dst = sgw$, and e^{d1}, e^{d2} are outgoing links of j. P and B: sets of working and protection paths, where $b_p \in B$ is link disjoint with working path $p \in P$

1: $P \leftarrow$ k shortest paths between j and sgw (Yen's Algorithm).
2: $B \leftarrow \emptyset$
3: **for** each path $p \in P$ **do**
4: Create a modified weights graph $\mathcal{G}''$ by making cost of links $e \in p$ a big number.
5: $b_p \leftarrow$Shortest path between j and sgw in $\mathcal{G}''$, to get a possible backup path of p
6: **while** $b_p \in B$ **do**
7: $b_p \leftarrow$ next shortest path (Dijkstra algorithm) between j and sgw in $\mathcal{G}''$, to get another possible backup path of p
8: **end while**
9: $B \leftarrow B \cup b_p$
10: **end for**
11: **return** (P, B)

resource allocation. A set of k candidate path pairs with 1+1 protection is pre-calculated for every demand in the network using Algorithm 1.

In order to cope with unforeseen traffic increase (e.g., due to flash-crowd events), both formulations reserve 10% of the wavelength channels ($\epsilon = 0.1$) in each active fiber to allow flexibility upon unexpected traffic demand increases; see (3.5e) and (3.6e). Upon exceeding an utilization threshold, a minimum-cost path algorithm is triggered. The set of costs used in such algorithm are: low cost for unassigned active wavelengths, medium cost for switched-off wavelengths on active fibers, and highest cost for wavelengths in switched-off fibers.

3.4.1 Virtual Wavelength Path (VWP)

A path formulation of VWP optimization problem is presented in equation (3.5). In VWP the OXCs have full wavelength conversion and fiber switching capabilities.

Each demand $d \in \mathcal{D}$ in formulation (3.5) requests h_d Gbit/s which can be split into different paths and provisioned with 1+1 protection (two link disjoint paths). In formulation (3.5) routing assignment per working and protection path pair is given by variable x_{rdp} [1 if r-th connection of demand d is routed on path pair (working and protection) p, zero otherwise]. The integer variables w_e and f_e represent active wavelengths and fibers. Pre-calculated path pairs are given by parameter δ_{edp} ($\delta_{edp} = 1$ if link e belongs to the path pair p realizing demand d, zero otherwise), which is obtained with the k-PLDP algorithm

proposed in Algorithm 1 of our previous work [208].

$$\text{Minimize } (\alpha + \mu + \gamma + \zeta) \sum_{e \in \mathcal{E}} f_e + \beta \sum_{e \in \mathcal{E}} w_e \tag{3.5a}$$

$$\textstyle\sum_{p \in \mathcal{P}_d} x_{rdp} = 1, \quad d \in \mathcal{D}\ r \in \mathcal{R}_d \tag{3.5b}$$

$$\textstyle\sum_{r \in \mathcal{R}_d} \sum_{d \in \mathcal{D}} \sum_{p \in \mathcal{P}_d} x_{rdp}\, \delta_{edp} \ \leq\ w_e, \quad e \in \mathcal{E} \tag{3.5c}$$

$$w_e \leq f_e W, \quad e \in \mathcal{E} \tag{3.5d}$$

$$f_e \leq (1 - \epsilon) F_e, \quad e \in \mathcal{E} \tag{3.5e}$$

$$x_{rdp} \text{ binary}, \ \ w_e \in Z^+, \ \ f_e \in Z^+ \tag{3.5f}$$

In VWP formulation, the objective (3.5a) is the minimization of the number of active fibers f_e and wavelengths w_e that are routing-dependent variables affecting the power consumption, as described in section 3.6.1.1. Solenoidality constraint (3.5b) enforces the provision of bandwidth to each connection request. Given the line rate of the wavelengths $L = 10$ Gbit/s, and maximum number of channels per fiber $W = 80$, expression (3.5c) determines the number of active wavelengths per link, and (3.5d) imposes an upper bound. Constraint (3.5e) limits the number of active fibers per link to F_e, and reserves some capacity (ϵ) to cover unexpected traffic demands. F_e is calculated in the static network-dimensioning phase, as presented in section 3.6.

3.4.2 Wavelength Path (WP)

We now describe the WP model which takes advantage of the small distances[3] in the network to establish fully transparent lightpaths, with optical bypass to avoid optical-to-electrical and electrical-to-optical (O/E-E/O) conversions in transit nodes. As a consequence, the transparent lightpaths have no wavelength conversion capabilities, and wavelength continuity constraint must be considered.

A path formulation of WP problem is presented in equation (3.6). WP formulation must perform routing and wavelength assignment (RWA) of working and protection paths using the binary variables $x_{drp\lambda}$ ($x_{drp\lambda} = 1$ if working path of r-th connection of demand d is routed on path p and wavelength λ, zero otherwise) and $y_{drp\lambda}$ ($y_{drp\lambda} = 1$ if protection path of r-th connection of demand d is routed on path p and wavelength λ, zero otherwise). The number of active wavelengths and fibers per link are represented by variables w_e and f_e respectively. The pre-calculated working paths are given by the parameter δ^w_{edp} ($\delta^w_{edp} = 1$ if link e belongs to the path pair p realizing working path of demand d on any wavelength, zero otherwise), and δ^p_{edp} ($\delta^p_{edp} = 1$ if link e belongs to the path pair p realizing protection path of demand d on any wavelength, zero otherwise) obtained with the k-PLDP algorithm (see Algorithm 1).

[3]Regeneration is normally needed after 1500 km for non-coherent wavelength channels at 10 Gbit/s [223].

$$\text{Minimize} \sum_{e\in\mathcal{E}}(\alpha+\mu+\gamma)f_e \tag{3.6a}$$

$$\textstyle\sum_{p\in\mathcal{P}_d}\sum_{\lambda\in\mathcal{W}} x_{drp\lambda} = 1, \quad d\in\mathcal{D},\ r\in\mathcal{R}_d \tag{3.6b}$$

$$\textstyle\sum_{\lambda\in\mathcal{W}} x_{drp\lambda} = \sum_{\lambda\in\mathcal{W}} y_{drp\lambda}, \quad d\in\mathcal{D},\ r\in\mathcal{R}_d,\ p\in\mathcal{P}_d \tag{3.6c}$$

$$\textstyle\sum_{d\in\mathcal{D}}\sum_{r\in\mathcal{R}_d}\sum_{p\in\mathcal{P}_d}(x_{drp\lambda}\delta^w_{edp} + y_{drp\lambda}\delta^p_{edp}) \le f_e, \quad e\in\mathcal{E},\ \lambda\in\mathcal{W} \tag{3.6d}$$

$$f_e \le (1-\epsilon)F_e, \quad e\in\mathcal{E} \tag{3.6e}$$

$$\textstyle\sum_{d\in\mathcal{D}}\sum_{r\in\mathcal{R}_d}\sum_{p\in\mathcal{P}_d}\sum_{\lambda\in\mathcal{W}} x_{drp\lambda}\delta^w_{edp} + y_{drp\lambda}\delta^p_{edp} = w_e, \quad e\in\mathcal{E} \tag{3.6f}$$

$$w_e \le f_e W, \quad e\in\mathcal{E} \tag{3.6g}$$

$$x_{drp\lambda}\ \text{binary},\ y_{drp\lambda}\ \text{binary},\ f_e\in R^+,\ w_e\in R^+ \tag{3.6h}$$

In formulation (3.6) the objective function (3.6a) minimizes the number of active fibers f_e, that is the only routing dependent variable that affects the power consumption at the optical layer of WP (as explained in section 3.6.1). The solenoidality constraint (3.6b) enforces the provision of each connection request. Constraint (3.6c) forces the utilization of a single path pair p to route the same request. Equations (3.6d) and (3.6e) determine and constrain the number of active fibers per link, reserving some capacity (ϵ) to allow unexpected traffic demand meeting the wavelengths continuity constraint. Equation (3.6f) determines the number of active wavelengths per link, that are constrained to $f_e W$ by (3.6g).

3.4.3 Complexity Analysis

Both WP and VWP are known to be NP-hard problems [64]. To give an idea of their complexity, we evaluated the number of constraints and variables of the proposed formulations (3.5) and (3.6). We indicate with R and K the average number of connection requests per each demand and the number of pairs of link disjoint paths, respectively. We can state that the number of constraints of VWP grows as $O(V(V-1)R)$, while for WP grows as $O(V(V-1)R(K+1))$. From the analysis of the variables introduced in the formulations, we can state that the number of variables in VWP grows as $O(V(V-1)RK)$, while WP grows as $O(V(V-1)RKW)$.

In order to simplify the problems, we have used path formulations to constrain the number of possible link disjoint path pairs to a given number (K). Using Yen's algorithm for k-shortest paths, and Dijkstra's algorithm implemented using a Fibonaci heap, the worst case complexity of Algorithm 1 is $O(k^2E(V+ElogE))$.

3.5 Online Dynamic Bandwidth allocation

The offline approach presented in section 3.4 assumes that there are regular traffic patterns in the mobile network. The predictability of traffic patterns allowed us to propose MILP models to find optimal routing (and wavelength) assignment solutions (RA and RWA). However, in the presence of unexpected variations over such regular traffic patterns, the offline approach may not be the best choice. Solving MILPs to get optimal solutions is time consuming and cannot be used for online decisions (fraction of second) upon unpredicted traffic variations. Heuristics are the common approach for online decision making. Algorithm 2 describes a proposed matheuristic for online RA or RWA (for VWP and WP respectively). Algorithm 2 is based on two phases:

- Phase 1 - Offline Planning Optimization: based on the MILP models of sections 3.4.1 and 3.4.2. Such models are solved for each $t \in \mathcal{T}$ using the predictable (regular) demand matrix ($h_{d,t}$) of an average weekday and weekend.
- Phase 2 - Online Routing: heuristic method that favors optimality by running a minimum cost algorithm on a set of optimally vertex-weighted graphs $\hat{\mathcal{G}}$. The optimal weights calculation is based on the optimal solution $\mathcal{S}_t$ obtained in phase 1.

Initially the set of reconfiguration time points $\mathcal{T}$ is composed by every hour of the day. However, hourly reconfigurations ($|\mathcal{T}| = 24$) is not well accepted by service operators because it leads to service disruption and instability of distributed routing algorithms, such as OSPF and EIGRP. Therefore in subsection 3.5.4 we present a simple scheduling heuristic to reduce the reconfiguration time points $|\mathcal{T}| < 24$.

Due to the wavelength assignment of the WP model 3.5.2, a layered graph $\mathcal{G}'$ is needed to perform RWA with a minimum-cost path algorithm. Thus, in subsection 3.5.2.1 we present the vertex-weighted layered graph $\hat{\mathcal{G}}'$, and subsection 3.5.3 describes the proposed modification of the well-known pair of edge disjoint path algorithm for $\hat{\mathcal{G}}'$.

3.5.1 Optimal weights for VWP model

For each $\hat{r} \in \{1..\hat{r}_{d,t}\}$ belonging to demand d at reconfiguration point t, a set of weights $\mathcal{C}_{d,t}$ is generated. Table 3.1 summarizes the 8 possible link weights $c_{e,\hat{r},d,t}$ that can be assigned to the vertex-weighted graph $\hat{\mathcal{G}}$.

When comparing the regular traffic demand $h_{d,t}$ with the short-term prediction[4] $\hat{h}_{d,t}$, there are two type of requests:

[4]In this article we do not focus on the short-term prediction methods. Thus we assumed to have a perfect prediction method that basically provides the precise traffic demand of the next reconfiguration time point. We refer the reader to [224] for a comparison on traffic prediction methods.

Algorithm 2 Online Routing Matheuristic

Solve off-line optimization models using the predictable demand matrix, and generate on-line optimal weights to reduce the routing problem to a Minimum cost algorithm

Phase 1 - Off-line Planning Optimization

Given historical traffic data-set $h_{d,t}$ (large observation windows)

for $d \in \mathcal{D}$ **do** ▷ $\mathcal{D}$ *Set of demands*

  for $t \in \{1, 2, ..., 23, 24\}$ **do** ▷ *One reconfiguration per hour*

    Get $h_{d,t}$: the regular (predictable) aggregated traffic demand of an average weekday or weekend (see section 3.3)

  end for

end for

Compute $\mathcal{T}$: set of reconfiguration time points which can be:
Hourly ($|\mathcal{T}| = 24$) or Scheduled ($|\mathcal{T}| < 24$) (in section 3.5.4 we present a reconfiguration time points scheduling algorithm)

for $t \in \mathcal{T}$ **do**

  Solve Optimization problem, get optimal solution $\mathcal{S}_t$ using:
  VWP (section 3.4.1) or WP (section 3.4.2)

end for

Phase 2 - On-line Routing

for $t \in \mathcal{T}$ **do**

  for $d \in \mathcal{D}$ **do**

    Step 2.1 - Short-time traffic prediction

    Given long-term prediction $h_{d,t}$ and the current traffic demand $\bar{h}_{d,t}$
    Predict $\hat{h}_d$: the short-term traffic demand for the next reconfiguration point.

    Step 2.2 - Optimal weights computation

    Given optimal $\mathcal{S}$ and the relation between $h_{d,t}$ and $\hat{h}_{d,t}$
    Compute $\mathcal{C}$: the weights of the optimal-weighted graph for each connection request $\hat{r}_{d,t} = \lceil \hat{h}_{d,t}/L \rceil$, where L is the line rate (see sections 3.5.1 and 3.5.2).

    for $\hat{r} \in$ {Real-time requests belonging to d at t} **do**

      Step 2.3 - Minimum cost algorithm

      Given the optimal-weighted graph of step 2
      Compute routing using a greedy algorithm based on Bhandari for VWP or on its modification (sec. 3.5.3) for WP.

    end for

  end for

end for

- Predictable ($\hat{r}_{d,t} \leq r_{d,t}$): all requests $\hat{r}$ are optimally planed.
- Unpredictable ($\hat{r}_{d,t} > r_{d,t}$): a sub-set of the requests are optimally planed $\{\hat{r} | \hat{r} \leq r_{d,t}\}$, while another subset is unexpected $\{\hat{r} | \hat{r} > r_{d,t}\}$.

Depending on the offline optimal planning results, for each request $\hat{r} \in \{1..\hat{r}_{d,t}\}$ there are 4 possible conditions of links $e \in \mathcal{E}$:

- Assigned: e belongs to the $\hat{r}$-th pair of working and backup paths.

Table 3.1: Optimal Link Weights ($c_{e,\hat{r},d,t}$) for VWP

Given the $\hat{r}$-th connection request of demand d at time t

Edge e **condition**	**Predicted** $\hat{r}_{d,t} \leq r_{d,t}$ $c_{e,\hat{r},d,t} =$	**Unpredict.** $\hat{r}_{d,t} > r_{d,t}$ $c_{e,\hat{r},d,t} =$
Assigned	1	Not possible
Available	$(\omega + 1)$	σ
Available-inactive	$(\omega + 1)\Delta$	Δ
Unavailable	$\mathcal{M}$	$\mathcal{M}$
$\mathcal{M}$: *big M*, $\sigma << 0$, ω = Length of backup path of $(\hat{r}, d, t)$ tuple		
Δ: *fiber-to-wavelength activation cost ratio* ($\Delta > 0$)		

- Available: e has at least one free wavelength (not assigned to any request or demand) in active fibers.
- Available-inactive: there is at least one inactive fiber of e.
- Unavailable: e has no free wavelengths.

3.5.2 Optimal weights for WP model

In this work we use an equivalent weighted-layered-graph transformation $\hat{\mathcal{G}}'$ in order to reduce the RWA with 1+1 protection to finding a pair of edge-disjoint paths in $\hat{\mathcal{G}}'$, at step 2.3 of the matheuristic presented in Algorithm 2. Before describing the optimal weighted-graph computation for WP models, we first characterize the equivalent layered-graph as proposed by Chen *et. al.,* in [225].

3.5.2.1 The layered-graph

The equivalent layered-graph transformation allows to perform RWA with a minimum cost algorithm (step 3 of the online routing for WP). Given the graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ (defined in section 3.4), it can be represented by its equivalent layered-graph $\mathcal{G}'(\mathcal{V}', \mathcal{E}', \mathcal{V}^s, \mathcal{A}^s, \mathcal{V}^d, \mathcal{A}^d)$. In $\mathcal{G}'$, each node $\mathcal{V}$ and link $\mathcal{E}$ in the original graph $\mathcal{G}$ is replicated WF times as virtual nodes $\mathcal{V}'$ and virtual links $\mathcal{E}'$. The virtual nodes of node $v \in \mathcal{V}$ are denoted as $v^{\lambda f}$, with wavelengths $\lambda \in \{1..W\}$ and fibers $f \in \{1..F\}$. The virtual links of link $e \in \mathcal{E}$ are denoted as $e^{\lambda f}$, with $\lambda \in \{1..W\}$ and $f \in \{1..F\}$. For each node $v \in \mathcal{V}$ in the original graph two extra dummy nodes are added, where source and destination of the demands $\mathcal{D}$ are mapped. The dummy source node related to node v, denoted as $v^s \in \mathcal{V}^s$, only has outgoing dummy-arcs $\mathcal{A}^s$ towards the virtual replicas of node v ($v^{\lambda f}$). The dummy destination node $v^d \in \mathcal{V}^d$, only has incoming dummy-arcs $\mathcal{A}^d$ from the virtual replicas of node v ($v^{\lambda f}$).

Fig. 3.6 depicts an example of $\mathcal{G}'$ for a 5 nodes network (on the top left of the figure), with $F = 2$ fibers per link and $W = 2$ wavelengths per fiber. In the example, wavelength conversion is possible at nodes 3 and 5, while fiber

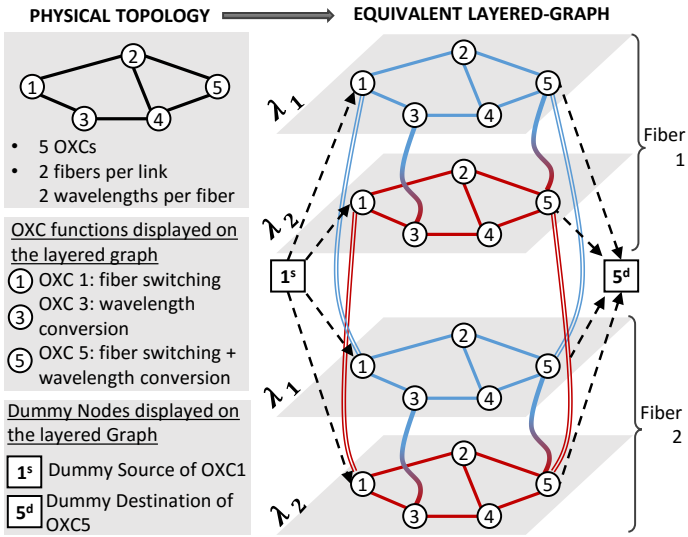

Figure 3.6: Equivalent layered-graph for the physical topology shown in the top left corner, and considering only two dummy nodes $1^s \in \mathcal{V}^s$ and $5^d \in \mathcal{V}^d$ where a demand from OXC1 to OXC5 can be mapped.

switching is enabled at nodes 1 and 5. To simplify the presentation of Fig. 3.6, only two dummy nodes with its corresponding 4 directed dummy-arcs are depicted, even if there are in total 10 dummy nodes and 20 outgoing and 20 incoming dummy-arcs in this network.

Based on the layered graph, the RWA problem with 1+1 protection can be reduced to finding a pair of edge-disjoint paths in $\hat{\mathcal{G}}'$ for the set of demands $\mathcal{D}$.

3.5.2.2 Optimal Weights for WP in the layered graph

For each $\hat{r} \in \{1..\hat{r}_{d,t}\}$ belonging to demand d at the next reconfiguration point t, a set of weights $\mathcal{C}'_{d,t}$ is generated. Table 3.2 summarizes the 10 possible weights $c'_{e',\hat{r},d}$ that can be assigned to the virtual links ($e^{\lambda f}$ will be called e' for simplicity) of the vertex-weighted layered-graph $\hat{\mathcal{G}}'$. As described in section 3.5.1, there are two type of requests: predictable and unpredictable demands. Depending on the offline optimal planning results S_t (phase 1 of Algorithm 2), there are 5 possible conditions of links $e' \in \mathcal{E}'$:

- Working: e' belongs to the $\hat{r}$-th working path of d.
- Backup: e' belongs to the $\hat{r}$-th backup path of demand d.
- Available: e' is a free wavelength of an active fiber.
- Inactive: e' belongs to an inactive fiber of e.

Table 3.2: Optimal Virtual Link Weights ($c_{e',\hat{r},d}$) for WP

Given the $\hat{r}$-th connection request of demand d at time t

Virtual Link e' **condition**	**Predicted** $\hat{r}_{d,t} \leq r_{d,t}$ $c_{e',\hat{r},d,t} =$	**Unpredict.** $\hat{r}_{d,t} > r_{d,t}$ $c_{e',\hat{r},d,t} =$
Working	σ	Not possible
Backup	1	Not possible
Available	$(\omega + 1)$	σ
Inactive	$(\omega + 1)\Delta$	Δ
Unavailable	$\mathcal{M}$	$\mathcal{M}$

$\mathcal{M}$: *big M*, $\sigma << 0$, ω = Length of backup path of $(\hat{r}, d, t, i)$ tuple
Δ: *fiber-to-wavelength activation cost ratio* ($\Delta > 0$)

Algorithm 3 Pair of link disjoint path over layered graph: PLDP-LG(src,dst)

1: Compute shortest path P_1 in $\hat{\mathcal{G}}'$ using Dijkstra algorithm
2: Set the weight of each virtual arc $a' \in P_1$ and all other virtual arcs related to the same physical path to a large number $c'_a = \mathcal{M}$
3: Negate the cost of the opposite directed arcs of P_1, such modified graph is called $\hat{\mathcal{G}}''$
4: Compute the shortest path P_2 using a modified Dijkstra or a BFS algorithm in $\hat{\mathcal{G}}''$
5: Build a reduced graph $\hat{\mathcal{G}}^{reduced}$ with paths P_1 and P_2, erase the interlacing links between P_1 and P_2
6: Given $\mathcal{G}^{reduced}$ use the simple two step approach to get the shortest pair of link disjoint paths

- Unavailable: e' is assigned to other requests $(r'|r' \neq \hat{r})$, or to other demands $(d'|d' \neq d)$.

In the layered graph $\hat{\mathcal{G}}'$, the cost of dummy arcs is always equal to zero $c_a = 0$.

3.5.3 Link-disjoint path-pairs computation

After computation of weights, an algorithm is used to compute the pair of link-disjoint paths to assign to protected request $\hat{r} \in \{1..\hat{r}_{d,t}\}$ belonging to demand $d \in \mathcal{D}$ at time t.

Each link $e \in \mathcal{E}$ belonging to the original graph $\mathcal{G}$ is replicated WF times as virtual links $e' \in \mathcal{E}'$ in the layered graph $\mathcal{G}'$. Therefore, virtual-edge disjointness on $\mathcal{G}'$ do not guarantee physical edge disjointness on $\mathcal{G}$. Thus, we have implemented a minimum cost greedy algorithm based on Bhandari's link-disjoint path algorithm [226]. Algorithm 3 describes the implemented extension of Bhandari's algorithm to enforce physical edge disjointness in a weighted layered graph (here after called modified Bhandari algorithm).

3.5.4 Reconfiguration Time Points Scheduling

Dynamic bandwidth allocation techniques reconfigure the network in relation to traffic demand variations. We considered a tidal traffic matrix, with per hour variations. Thus, the maximum resource allocation efficiency η defined by equation (3.7), is met by scheduling hourly reconfigurations $\mathcal{T} = \{1, 2, ..24\}$.

$$\eta = \frac{Total\_Demanded\_Bandwidth}{Total\_Allocated\_Bandwidth} \tag{3.7}$$

However, hourly reconfigurations are not well accepted by service operators. A solution is to reduce the reconfiguration time points ($|\mathcal{T}| < 24$) by scheduling reconfiguration events into specific time points $t \in \mathcal{T}$, creating a trade-off: a decrease of the reconfiguration time points $|\mathcal{T}|$ produce an increase of bandwidth over-provisioning (in consequence, increasing power consumption).

Given the traffic matrix of an specific day and lower threshold of the bandwidth allocation efficiency (expected efficiency $\bar{\eta}$), Algorithm 4 describes a *Simulated Annealing*-based heuristic method for optimizing the reconfiguration scheduling ($\mathcal{T}$) by finding the minimum number of reconfiguration time points ($|\mathcal{T}|$) that meet the expected allocation efficiency ($\eta \geq \bar{\eta}$).

3.5.5 Complexity Analysis

In the worst case, Algorithm 2 performs bandwidth allocation for each hour of the day ($|\mathcal{T}| = 24$). The complexity of Algorithm 2 is broken into two components: off-line and on-line phases.

Phase 1 - Off-line Planning Optimization: in section 3.4.3 we gave an idea of VWP and WP problems complexity. This models must be solved at each hour of the day $|\mathcal{T}| = 24$ using the predictable traffic demand patterns ($h_{d,t}$).

Phase 2 - On-line heuristic routing: At each reconfiguration time point, and for each demand the on-line component of the Algorithm 2 must perform: short-time traffic prediction, optimal weights computation and link-disjoint path-pair computation. The worst case complexity of the heuristic is dominated by the calculation of the optimal weights: *VWP* grows with $O(V^2(V-1)^2RE)$; while *WP* grows with $O(V^2(V-1)^2REWF)$, where W and F are the maximum number of wavelengths channels and fibers, respectively.

Algorithm 4 Reconfiguration time point Scheduling $(\bar{\eta}, H)$

1: $\bar{T} \leftarrow 0$ ▷ $\bar{T}$: maximum number of reconfigurations
2: **while** $\eta < \bar{\eta}$ **do** ▷ $\bar{\eta}$: expected efficiency, see equation (3.7)
3: $\bar{T} \leftarrow \bar{T} + 1$ ▷ $\mathcal{T}$: set of reconfiguration time points
4: **do** *Simulated Annealing* to Find reconfiguration schedule $\mathcal{T}$ such that $|\mathcal{T}| == \bar{T}$ and with maximum efficiency η
5: **end** **while**
return $\mathcal{T}$, the reconfiguration scheduling with min. number of reconfiguration time points and expected allocation efficiency

3.6 Results

3.6.1 Power Consumption Models

In this work we only considered the power consumption of optical layer. Power data of components are based on the models given in [223], that were used to set the weights in the objective functions of equations (3.5) for VWP and (3.6) for WP [5].

3.6.1.1 Power consumption of VWP (full grooming scenario)

The optical layer power dissipation of VWP can be obtained by (3.8).

$$P^{VWP} = \psi V + \beta \sum_{e \in \mathcal{E}} w_e + (\alpha + \mu + \gamma + \zeta) \sum_{e \in \mathcal{E}} f_e \quad [W] \tag{3.8}$$

where V is the number of OXCs (one per CN) in the network, with a fix consumption of $\psi = 150$ W per OXC. ψV is the only routing independent term in (3.8). In VWP each wavelength is terminated at every node to perform traffic grooming, thus we add $\beta \sum_{e \in \mathcal{E}} w_e$ in equation (3.8).

For a line-rate of $L = 10$ Gbit/s, the power dissipation of transponder/-muxponder per active wavelength (in one direction) is $\beta = 25$ W. The cost per active fiber (f_e) in the OXCs is $\gamma = 85$ W (optical switching), and $\zeta = 50$ W (add/drop ports). A WDM terminal consumes $\mu = 120$ W (two terminals are installed per fiber). The optical line amplifier (OLA) dissipates $\alpha = 32.5$ W per active fiber.

3.6.1.2 Power Consumption of WP (optical bypass scenario)

The power dissipation of WP is given by (3.9).

$$P^{WP} = \psi V + (\beta + \zeta) \sum_{d \in D} r_d + (\alpha + \mu + \gamma) \sum_{e \in \mathcal{E}} f_e \quad [W] \tag{3.9}$$

In WP each connection request is satisfied by two transparent link-disjoint lightpaths from source to destination, therefore the contribution of transponder/muxponder β W and add/drop fiber ports ζ W are independent of the routing. Accordingly, the only routing dependent variable that affects the power consumption is f_e, with a cost of $\alpha + \mu + \gamma$ W.

3.6.2 Offline Numerical results

Figs. 3.7 and 3.8 depicts the results of total power consumed (kW) for three different approaches of VWP and WP models, respectively. Thanks to regular traffic patterns, all computations can be done off-line, once for a whole operational period (e.g. two months). The flat lines represent the static network

[5] Symbols $\mathcal{E}, e, w_e, f_e, \mathcal{D}, d, r_d$, were defined in sections 3.4 and 3.5.

configuration (VWP and WP - Static), where all the elements are on to cope with the peak hour demand of the historical data set. The step-like curves with changes every hour (VWP and WP - Hourly), correspond to minimization of active resources, by solving one instance of the optimization models per hour ($|\mathcal{T}| = 24$). The continuous curves with long steps represent the behavior of a network that schedules a reduced set of reconfigurations per day ($|\mathcal{T}| < 24$), using the heuristic method presented in subsection 3.5.4 (VWP and WP - Scheduled). At least 20% of energy dissipation E (kWh) per day can be saved using dynamic network operation in both WP and VWP cases.

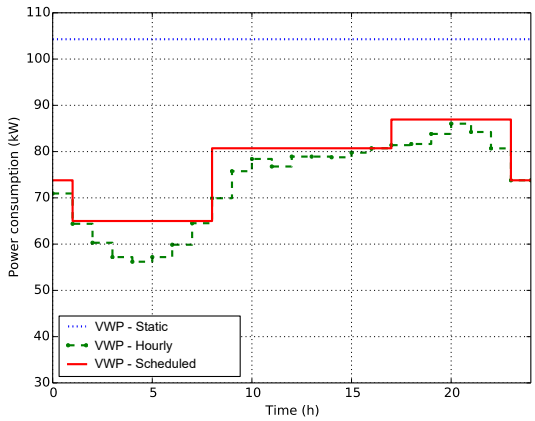

Figure 3.7: Optical layer power consumption of the mobile metro-core network under predictable aggregated traffic using three of-line resource allocation methods: Static-VWP, Hourly-VWP and Scheduled-VWP (Time point optimized).

Table 3.3: Energy dissipation and disruption rate trade-off

	VWP		WP	
	Hourly	Scheduled	Hourly	Scheduled
Total Energy (kWh)	1760	1850.6	1314	1358.3
Energy Saving (compared to static)	29.7 %	26.1 %	23 %	20 %
Average service disruption rate	49 %	11.9 %	95.8 %	16.7 %

Our results demonstrate that WP (reducing O/E-E/O conversions with optical bypass) is more energy efficient than VWP for mobile metro-core networks. WP consumes 22 % less energy E than VWP. However, table 3.3 shows that WP is more prone to service disruption than VWP. We considered that service disruption in VWP is experienced only for changes in the routing assignment (RA) of each request, while in the WP, due to wavelength continuity

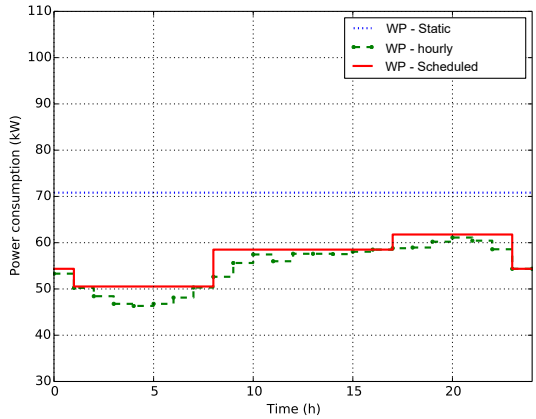

FIGURE 3.8: Optical layer power consumption of the mobile metro-core network under predictable aggregated traffic using three of-line resource allocation methods: Static-WP, Hourly-WP and Scheduled-WP (Time point optimized).

constraint, wavelength reassignments must also be taken into account, reflecting a great increase of the disruption rate defined by equation (3.10).

$$Average\ Disruption\ rate = \frac{Total\_disrupted\_requests}{Total\_requests} \tag{3.10}$$

Table 3.3 summarizes the trade-off between energy dissipation E kWh, and service disruption when using dynamic resource allocation. Our results prove that by properly scheduling the reconfigurations time points it is possible to considerably reduce service disruption rate (WP from 95.8% to 16.7% and VWP from 49% to 11.9%), with a small penalty on energy savings (WP from 23% to 20% and VWP from 29.7% to 26%).

The common approach in today's mobile metro-core networks is similar to the static VWP scenario ($E = 2502.9$ [kWh per day]), so it is possible to save up to 47.5% of energy per day using the proposed load adaptive operation with optical bypass WP ($E = 1314$ [kWh per day]).

3.6.3 Online results

In order to assess the performance of the proposed matheuristic, a Discrete event simulator (DES) was built using SimPy [227], a process-based DES framework based on Python. The on-line procedures were tested over one month of real-time metropolitan tidal traffic.

Figs. 3.9 and 3.10 provide a comparison of the proposed on-line matheuristic (described by Algorithm 2) with the off-line optimization, and simple on-line heuristic (based on minimum cost algorithm using number of hops as link weights) for VWP and WP, respectively.

Table 3.4, summarizes the percentage of daily energy dissipation savings when using: off-line optimization, matheuristic and heuristic methods.

The results demonstrate the effectiveness of the proposed matheuristic, with an optimality gap below 1.5% in the average energy dissipation, while the simple heuristic displays an optimality gap of almost 10%.

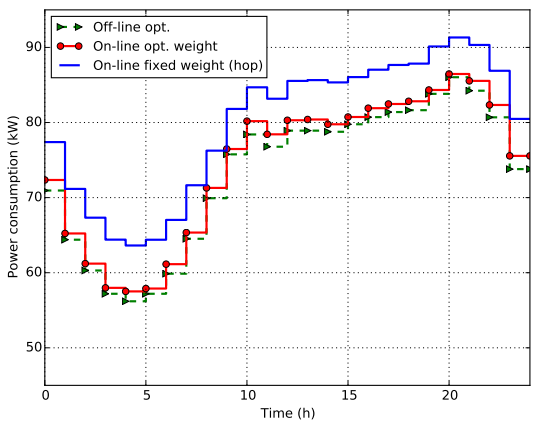

Figure 3.9: Energy dissipation in the mobile metro-core network for three different methods: VWP off-line Optimization (under predictable aggregated traffic), VWP on-line matheuristic and VWP on-line heuristic.

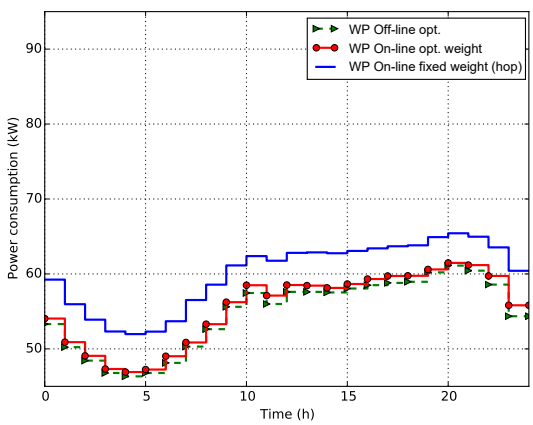

Figure 3.10: Energy dissipation in the mobile metro-core network for three different methods: WP off-line Optimization (under predictable aggregated traffic), WP on-line matheuristic and WP on-line heuristic.

Table 3.4: On-line results - Average Energy consumption increase

	On-Line Matheuristic	On-line Heuristic (Fixed weight)
VWP	1.5 %	8.9 %
WP	1.3 %	9.7 %

3.7 Concluding Remarks

In this Chapter, we analyzed the impact of user mobility patterns on the aggregated tidal traffic effect that is offered to a mobile metro-core network. Using real user-mobility data and an aggregation network architecture built from real cell site locations, we showed that the high predictability of regular human movements actually creates predictable spatio-temporal fluctuations at the aggregated traffic of the metro-core. Moreover, human mobility patterns are related to the social functions of the different metro areas, which provides valuable hints on the expected trend of the aggregated traffic.

By knowing the predictable traffic demands, we proposed two off-line optimization models for dynamic resource allocation and reported daily energy savings of more than 20% while effectively responding to traffic load variations.

The smaller energy dissipation was achieved by the dynamic WP model (section 3.4.2) with savings up to 47.5% when compare with static VWP, showing the advantages of load adaptive network operation and optical bypass, that thanks to the small link length of a mobile metro-core network can completely avoid optical regeneration.

The on-line approach manage unpredictable traffic components alongside with the predictable components, with a small optimality gap.

The scheduling procedure is an interesting trade-off between energy efficiency and service disruption. It allows to optimize the reconfiguration time points, in order to reduce disruption while keeping energy efficiency over an specified threshold.

This is an initial study for green mobile metro-core networks. We are now developing a more sophisticated VWP model that includes both traffic grooming and optical bypass based on traffic conditions, which promise to achieve better results.

Differential delay constrained multipath routing for SDN and optical networks 4

Historically, Internet traffic has been routed over the shortest path: which was convenient for best-effort data traffic; but it is not always suitable for today's scenario. Multipath (MP) routing is an effective technique for applications imposing stringent requirements on bandwidth, delay and availability. MP routing overcomes few limitations of single path routing like increased throughput and network resource utilization and resilience to possible link failures. However, different data paths over the network face different end-to-end path delay which is called differential delay DD. DD limits the performance gain of MP routing.

In MP routing, maximization of the cardinality K of the disjoint-path set for a given source and destination assuming an upper bound on the differential delay DD is one of the key factors enabling its practical applications. In this Chapter and in Chapter 5 we focus on differential delay mitigation techniques for MP optical routing. More specifically, in this chapter we study such an optimization problem for multipath routing involving maximization of K under the DD constraint as the primary objective, and then minimization of the average end-to-end transfer delay for the fixed (maximum) K under the same DD constraint. The optimization approach is iterative, based on solving an inner mixed-integer programming subproblem to minimize the delay for a given value of K and DD. In order to increase the solution space, we consider the strategy of allowing controlled routing loops. Such a technique is implementable in SDN and optical networks. We present numerical results illustrating the gain achieved by using controlled loops in comparison with the traditional loop-free approach.

4.1 Introduction

Multipath (MP) routing is a network functionality controlling splitting of the data flow from a source to a destination among multiple physical paths, and reconstructing it at the destination as a single flow, prior to being delivered to an upper layer.

Today's networks generally offer MP capability, e.g., in case of multi-homed datacenters, or end-devices with multiple interfaces, and multiple gateways for enterprise and carrier networks. However, the common networking practice seldom exploits MP.

From the viewpoint of network-user experience, multihoming is important for many reasons. Being connected to several carriers provides a way to improve the resiliency of network connectivity to failures and outages. Multihoming also allows a user to choose the provider offering best techno-economic conditions. Enterprises and content providers can achieve significant benefits in terms of resiliency and performance by multihoming [228].

Multihoming is very important in the context of mobile networks, which are rapidly gaining prominent importance, due to the exponential increase of mobile data traffic. Today's smartphones are equipped with both WiFi and 3G/4G interfaces and users often have access to multiple communication channels simultaneously. Users expect their connections to stay active even when their smartphones switch from one wireless network (e.g. WiFi) to another (e.g. LTE). To get increased performance, and in consonance with 5G network requirements, the mobile host must be multi-homed and use bandwidth aggregation by striping the data across multiple interfaces.

Regarding optical networks MP routing allows 1) meeting stringent requirements on bandwidth and delay [27, 229], 2) improving network resource utilization, load balancing and protection techniques [30, 230], and 3) minimizing the impact of SCC and spectrum fragmentation in flexi-grid networks [231].

In SDH networks, for instance, the MP VCAT (virtual concatenation) technique allows for a better utilization of network resources [27]. In optical transport networks (OTN), MP can be exploited to drastically decrease the amount of bandwidth reserved for protection [230].

In another context, MP is used by the Multipath Transport Control Protocol (MPTCP) [232] to augment the throughput of the TCP-based applications in a transparent way, i.e., without modifying the applications and yet preserving backward compatibility with TCP. In Chapter 7, we present the design of an SDN-based path manager for MPTCP and the implementation of an automated testbed for performance evaluation.

In conclusion, multipath routing significantly improve performance and resiliency of current networks for different kinds of scenarios, like data centers, enterprise networks, carriers and mobile hosts.

But why the common networking practice seldom exploits MP routing? One of the major obstacles is the data reconstruction operation at the destination: if

significant differences in the delays occur between the paths of the MP set, the reconstruction buffer has to be increased in size and the reordering task becomes time consuming. Consequently, the quality of the service delivered over the MP connection starts degrading, causing users experiencing unacceptable waiting times to receive the data (e.g., in conversational or gaming applications) or – even worse – an unexpected bursty-mode operation (e.g., in video streaming).

Another issue is related to routing: MP improvements in terms of throughput, load balancing, reliability and protection bandwidth are all fully achievable only provided that the paths of the MP are link disjoint. For example, it was proven that the TCP performance can be enhanced by MPTCP only if physical path-disjointness is enforced [233]. In a bandwidth constraint network the throughput of the MP connection can still increase as the number of paths K grows, while this is not possible for single path routing.

4.2 Related works

The problem of Differential Delay Routing (DDR) was first studied by [28] for the Ethernet over SONET architectures. The DDR problem was defined as follows: given a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, find K paths of unit capacity from source S to destination T such that their differential delay DD is upper bounded by a given constant Δ. DD is defined as the vector containing the differences in delay between the longer and all the other paths belonging to the MP set.

Srivastava *et al.* proved in [27] that the DDR problem is NP-hard, and that it is as hard as the longest path problem. Due to such complexity, the authors of [27] presented heuristic algorithms to solve the DDR problem by transforming it into a flow maximization problem over a set of paths that meet the DD constraint ($D \leq \Delta$).

Link disjointness was not considered in the DDR-problem [27][28]. The work presented by Sheng *et al.* [230] exploits the DDR-problem for shared path protection schemes; therefore, it was the first work to include link disjointness as a constraint in a DDR-related problem to reduce the required bandwidth allocated for protection. The authors of [230] relied on heuristics to solve the problem, as DDR (and consequently DDR with link disjointness) is NP-hard.

In [231], inverse routing was considered in OTN over GMPLS-controlled Flexi-grid networks in order to minimize the impact of SCC and spectrum fragmentation of elastic optical networks. Due to the reconstruction buffer requirements, the authors of [234] addressed the differential delay aware version of the problem presented in [231]. No DDC technique was used in [231] or [234].

4.2.1 Contribution

In [29] we extend the previous work by redefining the problem as follows: (a) find a set of K link-disjoint paths from source S to destination T that

maximizes K such that DD is upper-bounded by a given value Δ; (b) find the set of K^0 link-disjoint S-T paths minimizing the average end-to-end delay L (where K^0 is the maximum found in phase (a)) under the same DD constraint. We call this the 3D problem (Disjoint-Differential-Delay).

In fact, when looking for maximum K in phase (a) we could assume some fixed upper bound on L as well. Otherwise, looking for a large cardinality set of multiple paths with similar delays could easily result in selecting very long paths. This would have a negative impact on the quality of service, as it would slow down the applications (especially when automatic congestion control is adopted, as with TCP or MPTCP), and at the same time would increase network congestion.

To reduce the complexity of the 3D problem, we will adopt an iterative procedure involving a subproblem solved by means of a mixed-integer linear programming (MILP) formulation, as described in Section 4.3.

In general, enforcement of the DD constraint in the 3D problem can result in appearance of routing loops associated with the paths. Such cycles are usually undesired in networking but in the MP context they can compensate for delay differences without adding buffering capabilities at the destination nodes or transit nodes of a path, as proposed in [30].

Some network technologies, such as software defined networks [235], MPLS and optical networks, are potentially able to handle routing loops without generating "broadcast storms" (infinite loops) or routing failures. Therefore, a 3D formulation that allows for loops is also presented.

The comparison of the numerical results obtained by the loop-free model (3D-LF) with the results of the model which allows loops (3D-LIP) exhibits an advantage of 3D-LIP. This justifies the interest for the 3D-LIP case. In the Chapter we focus on the optimization aspects and do not discuss practical implementation of 3D-LIP in real networks.

4.3 Optimization models

Consider a network represented by an undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ are the sets of nodes and (undirected) links, respectively. Each link $e \in \mathcal{E}$ is associated with two oppositely directed arcs e' and e'' joining its nodes. The set of all such arcs is denoted by $\mathcal{A}$ (note that the resulting directed graph $\mathcal{G}(\mathcal{V}, \mathcal{A})$ is bi-directed). The source and destination nodes of arc a will be denoted by $s(a)$ and $t(a)$, respectively. The weight w_a of arc $a \in \mathcal{A}$ represents its delay. The sets of arcs $\delta^+(v)$ and $\delta^-(v)$ represent the outgoing star of arcs from node $v \in \mathcal{V}$ and the incoming star of arcs directed to v, respectively.

Problem 3D consists in finding an MP connection between a given source S and destination T node pair in graph $\mathcal{G}$ that complies with given differential delay Δ and average end-to-end delay Λ upper bounds, and as first objective maximizes the number of disjoint paths K and then minimizes the average end-to-end delay L.

Algorithm 5 Iterative procedure

Given differential delay Δ and average end-to-end delay Λ upper bounds: find the solution $\mathcal{S}_k$ that maximizes k constraint to Δ and Λ

```
k ← 2
while 1 do
    Get S_k by solving the optimization problem 3D* with k as an input parameter
    if No solution is found or the optimal L is greater than Λ then
        Break
    else
        k = k + 1
    end if
end while
if K == 2 then
    Reject the MP connection
else
    Return {S_{k-1}, k − 1}
end if
```

The MP connection is a set of K link-disjoint paths $\mathcal{P}$ from S to T in the directed graph $\mathcal{G}(\mathcal{V}, \mathcal{A})$. The paths in $\mathcal{P}$ are denoted by $P_k, k \in \mathcal{K}$, where $\mathcal{K} = \{1, 2, \ldots, K\}$. The delay W_k experienced by the k^{th} path in $\mathcal{P}$ is given by $W_k = \sum_{a \in P_k} w_a,\ k \in \mathcal{K}$. The average end-to-end delay of the MP connection $\mathcal{P}$ is given by $L = \frac{1}{K} \sum_{k \in \mathcal{K}} W_k$.

In order to simplify the formulation and to make 3D problem tractable, we introduce an iterative procedure in Algorithm 5 for maximizing K for a given value of Δ.

Consecutive instances of problem 3D* (see below) are solved for consecutive increasing K (starting with $K = 2$) until the problem becomes infeasible.

In the following subsections 4.3.1 and 4.3.2, two formulation of the optimization (sub)problem 3D* are presented. An MP connection request is described by the quadruple (S, T, Δ, K), where S, T are the source and destination nodes, Δ is the differential delay upper bound, and K is the assumed number of disjoint paths.

4.3.1 3D*-Loop-Free formulation (3D*-LF)

A node-link formulation of the 3D* problem is given in (4.1). The formulation uses binary arc-flow variables x_{ak} ($x_{ak} = 1$ if arc $a \in \mathcal{A}$ is used by path P_k, $k \in \mathcal{K}$; 0, otherwise) and the nonnegative variable h_{vk} (h_{vk} is the hop count from the source to a node $v \in \mathcal{V}$ for each path P_k, $k \in \mathcal{K}$). In the formulation, objective (4.1a) minimizes the cumulative delay of the MP connection. The flow conservation constraint (4.1b) makes the flows x_{ak} represent K paths from source S to destination T. Constraint (4.1c) forces link disjointness of the paths. The differential delay constraint is formulated in (4.1d)-(4.1e).

The last constraint (including a "big M" parameter) (4.1f) ensures that loops, otherwise advantageous in optimal solutions from the viewpoint of (4.1d)-(4.1e), cannot appear. In consequence, any feasible solution of (4.1) will be loop-free. Formulation (4.1) will be called 3D*-loop-free (3D*-LF in short).

$$\textbf{3D*-LF:}\ \text{minimize}\ \sum_{a\in\mathcal{A}}\sum_{k\in\mathcal{K}} w_a x_{ak} \tag{4.1a}$$

$$\sum_{a\in\delta^+(v)} x_{ak} - \sum_{a\in\delta^-(v)} x_{ak} = \begin{cases} 0, & v\in\mathcal{V}\setminus\{S,T\} \\ 1, & v=S \\ -1, & v=T \end{cases}, \quad v\in\mathcal{V}, k\in\mathcal{K} \tag{4.1b}$$

$$\sum_{k\in\mathcal{K}} x_{e'k} + x_{e''k} \le 1, \quad e\in\mathcal{E} \tag{4.1c}$$

$$\sum_{a\in\mathcal{A}} w_a x_{ak} - \sum_{a\in\mathcal{A}} w_a x_{am} \le \Delta, \quad k,m\in\mathcal{K}, k<m \tag{4.1d}$$

$$\sum_{a\in\mathcal{A}} w_a x_{am} - \sum_{a\in\mathcal{A}} w_a x_{ak} \le \Delta, \quad k,m\in\mathcal{K}, k<m \tag{4.1e}$$

$$h_{t(a)k} - h_{s(a)k} \ge 1 - M(1-x_{ak}), \quad a\in\mathcal{A}, k\in\mathcal{K} \tag{4.1f}$$

$$x_{ak}\ \text{binary},\ \ h_{vk}\ge 0. \tag{4.1g}$$

Note that K is an input parameter of the formulation, and not an objective.

4.3.2 3D*-Loops-in-path formulation (3D*-LIP)

Now we introduce an important and novel variation of formulation (4.1). As we have already mentioned, formulation (4.1) eliminates loops due to (4.1f), both in-path loops (like F-G-H in path S-F-G-H-F-T in Fig. 4.1) and isolated loops (disjoint with the main part of an $S-T$ path). Although the latter paths are clearly not allowed, the former could be effectively used to compensate path delays as required by the DD requirement. Thus, now we assume that the in-path routing cycles are allowed and manageable by the network in order to accomplish differential delay compensation. The appropriate formulation, called 3D*-LIP (loops-in-paths), is similar to the formulation of 3D*-LF (4.1), the differences are exposed in the following.

As 3D*-LF, the 3D*-LIP formulation uses binary flow variables x_{ak} to specify the path-set. Now, however, we use two extra sets of auxiliary nonnegative continuous variables. The first set, z_{avk} ($z_{avk}=1$ if arc $a\in\mathcal{A}$ is used by an artificial flow of value 1 from node $v\in\mathcal{V}$ on path $P_k, k\in\mathcal{K}$, to T; 0, otherwise).

3D-LIP			3D-LF		
Δ	K	L	Δ	K	L
0	2	12	0	Infeasible	
1	2	12	1	2	24,5

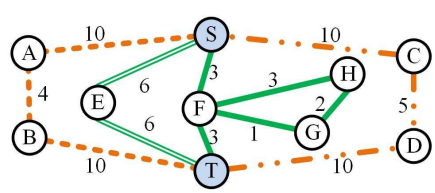

Figure 4.1: Comparison of 3D-LIP and 3D-LF in a 10-node network

The second set, r_{vk} ($r_{vk} = 1$ if node $v \in \mathcal{V}$ belongs to path P_k, $k \in \mathcal{K}$; 0, otherwise) is used to define the values of artificial flows.

The 3D*-LIP formulation shares the objective function (4.2a) and the first four constraints (4.2b)-(4.2e) with 3D*-LF (4.1). However, following [236], the loop avoidance constraint of 3D*-LF (4.1f) is exchanged by inequalities (4.2f)-(4.2g) and equalities (4.2h) in order to allow in-path loops but avoid isolated cycles. Artificial flow z_{avk} is initiated at each node $v \in \mathcal{V} \setminus \{S,T\}$ if $x_{ak} = 1$ and $a \in \delta^+(v)$. Constraints (4.2f), (4.2g) and (4.2h) define an artificial flow connecting node v to the destination node T. Clearly, equations (4.2h) represent the flow conservation constraints in a graph induced by the links in path P_k, i.e., in the graph $(\mathcal{V}, \{a \in \mathcal{A} : x_{ak} = 1\})$.

$$\textbf{3D*-LIP:}\ \text{minimize}\ \sum_{a\in\mathcal{A}} \sum_{k\in\mathcal{K}} w_a x_{ak} \tag{4.2a}$$

$$\sum_{a\in\delta^+(v)} x_{ak} - \sum_{a\in\delta^-(v)} x_{ak} = \begin{cases} 0, & v \in \mathcal{V} \setminus \{S,T\} \\ 1, & v = S \\ -1, & v = T \end{cases}, \quad v \in \mathcal{V}, k \in \mathcal{K} \tag{4.2b}$$

$$\sum_{k\in\mathcal{K}} x_{e'k} + x_{e''k} \leq 1, \quad e \in \mathcal{E} \tag{4.2c}$$

$$\sum_{a\in\mathcal{A}} w_a x_{ak} - \sum_{a\in\mathcal{A}} w_a x_{am} \leq \Delta, \quad k,m \in \mathcal{K}, k < m \tag{4.2d}$$

$$\sum_{a\in\mathcal{A}} w_a x_{am} - \sum_{a\in\mathcal{A}} w_a x_{ak} \leq \Delta, \quad k,m \in \mathcal{K}, k < m \tag{4.2e}$$

$$z_{avk} \leq x_{ak}, \quad a \in \mathcal{A}, v \in \mathcal{V} \setminus \{S,T\}, k \in \mathcal{K} \tag{4.2f}$$

$$r_{vk} \geq x_{ak}, \quad a \in \delta^+(v), v \in \mathcal{V} \setminus \{S,T\}, k \in \mathcal{K} \tag{4.2g}$$

$$\sum_{a\in\delta^+(u)} z_{avk} - \sum_{a\in\delta^-(u)} z_{avk} = \begin{cases} 0, & u \in \mathcal{V} \setminus \{v,T\} \\ r_{vk}, & u = v \end{cases},$$
$$u \in \mathcal{V} \setminus \{S\}, v \in \mathcal{V} \setminus \{S,T\}, k \in \mathcal{K} \tag{4.2h}$$

$$x_{ak}\ \text{binary},\ \ z_{avk}, r_{vk} \geq 0. \tag{4.2i}$$

4.4 Results

In this section we compare optimal solutions of 3D-LF and 3D-LIP in terms of the MP cardinality K and its average end-to-end delay L, considering Δ as a parameter. For simplifying the presentation, we do not impose any bound on L used in Step 4 of the procedure in Section 4.3.

The sample weighted graph shown in Fig. 4.1 illustrates how LIP can improve the 3D problem. For $\Delta = 0$, 3D-LF is infeasible, while 3D-LIP successfully finds two paths with equal delay by exploiting the LIP path S-F-G-H-F-T for delay compensation. For $\Delta = 1$ the optimum for both formulations is $K = 2$. However, the L of the solution provided by 3D-LIP is smaller.

Now we compare the two formulations on the 14-node network depicted in Fig. 4.2. The delay of each link w_e is random with uniform distribution

in $\{1, 2, \ldots, 5\}$. Fig. 4.3 and Fig. 4.4 show the results for $\Delta = 0, 1, \ldots, 6$, averaged over 100 instances of such random delay settings. (We have skipped confidence intervals as they do not influence the general picture.)

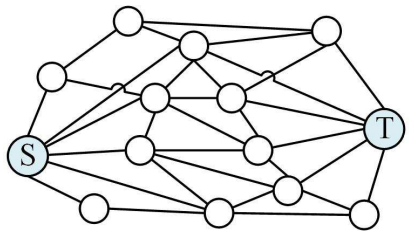

Figure 4.2: 14-node test network

As expected, for both formulations the cardinality of MP connection is strictly increasing with Δ. For the source S and destination T node pair in the network of Fig. 4.2 the maximum number of link-disjoint paths is equal to 6, imposing the upper bound for the maximization of K. Although the gain obtained by allowing LIP is not so significant, Fig. 4.3 shows how 3D-LIP improves the solution subset of Δ values. In particular, for $\Delta = [4, 5, 6]$, 3D-LIP always reaches the upper bound of $K = 6$ (while 3D-LF does not), with a small impact on L as compared with 3D-LF (see Fig. 4.4).

It can be noticed in Fig. 4.4 that L is almost always decreasing with Δ and that L is forced to considerably increase as the requirement on DD gets more stringent. In particular, maximum L is observed for $\Delta = 0$. This reveals a clear trade-off between delay minimization and delay equalization in MP communications. 3D-LIP(L) is slightly larger than 3D-LF(L) due to those cases when 3D-LIP finds more paths than 3D-LF.

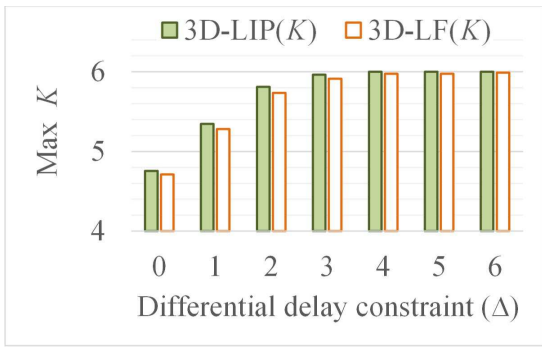

Figure 4.3: Comparison of 3D-LIP and 3D-LF in a 14-node test network based on Maximum K.

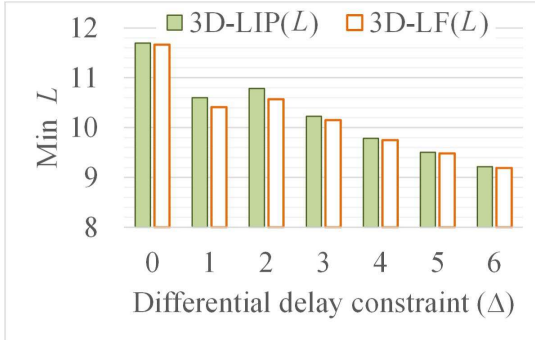

FIGURE 4.4: Comparison of 3D-LIP and 3D-LF in a 14-node test network, based on Minimum L.

4.5 Concluding Remarks

In this Chapter we have proposed an iterative optimization approach that simplifies the 3D (Disjoint-Differential-Delay) formulation and makes the problem tractable. We have shown how controlled in-path routing cycles broaden the solution space of the problem and can be used to equalize path delays as required by the DD constraint. The optimization procedure 3D-LIP that allows for such loops always finds better (or equally good) solutions than the traditional loop-free approach (3D-LF) when maximizing the number K of disjoint paths in order to increase the throughput of the MP connection. For both formulations there is a trade-off between the differential delay and the average end-to-end delay of the MP connection. This trade-off must be resolved, depending on the application, by proper setting of DD and L upper bounds.

This work represents an initial step in considering important issues of the MP networking, as we have focused only on the delay measures (DD and L) and assumed unlimited network capacity. In further steps, we will include such additional features as finite link capacity, delay variations over time, node disjointness, and concurrent routing of several MP demands. This additional features will make the problem even harder to solve.

The presented model can be applied to software defined, MPLS, and optical networks. For instance, an optimal routing solver implementing the described procedure could be implemented as a network App connected to the northbound interface of an SDN controller. Such kind of possible developments will be described in chapter 7.

Multipath Optical Routing with Compact Fiber Delay Line-based Differential Delay Compensation 5

In Chapter 4 we identified a trade-off between differential delay minimization and average end-to-end delay when using differential delay-constrained MP routing. Such trade-off led us to investigate differential delay-compensated MP routing for optical transport networks. In presence of differential delay the destination of a MP connection receives a disordered version of the original packet sequence. Differential delay compensation (DDC) techniques allow to recover the original sequence. DDC is normally performed at destination (centralized-DDC) using high speed reconstruction buffers. For MP connections with large DD the centralized-DDC creates a bottleneck that limits the performance gain of MP routing. DDC can be distributed along the paths (distributed electronic-DDC) to reduce the reconstruction buffer requirements and minimize DD at destination. In optical networks, distributed electronic-DDC incurs in extra costly and power hungry electro/optical (E/O) conversions, that are otherwise avoided by routing all optical circuits (i.e., lightpaths). To avoid extra E/O conversions, distributed electronic-DDC can be jointly placed with optical regeneration. Nonetheless, such approach greatly reduces the candidate nodes to distribute the DDC, because optical regeneration is only needed for very long lightpaths. This chapter proposes, for the first time, the use of compact fiber delay lines (FDL)s to perform distributed all optical DDC (transparent-DDC). The FDLs are passive elements that overcome the problems of previous solutions: they are not restricted to optical regeneration points, and do not incur into extra E/O conversions. An integer linear programming formulation is presented for the MP routing with DD-minimization problem that combines electronic-DDC co-located with 3R (Reamplifying, Reshaping and Retiming) regeneration points to the novel transparent-DDC based on

FDLs. Numerical results show the advantages of combining transparent and electronic-DDC in realistic network scenarios.

5.1 Introduction

The main problem of MP optical routing is that each path in an MP group experiences a specific delay, leading to a differential delay (DD) between the paths. In a MP connection, we define DD as the vector containing the differences in delay between the longer and all the other paths belonging to the MP set. In presence of DD, the destination receives a disordered version of the original packet sequence.

Fig. 5.1 depicts an MP connection composed by a MP set ($\mathcal{P}$) of 3 all optical paths $\mathcal{P} = \{p_1, p_2, p_3\}$; the longer path is p_1 and the corresponding DD between p_1 and p_2 is $DD_{1,2} = 1$ms, while DD between p_1 and p_3 is $DD_{1,3} = 0.5$ms. The differential delay vector is $DD = \{DD_{1,2}, DD_{1,3}\}$. In order to recover the original signal, the sub-channels must be synchronized by compensating $DD_{1,2}$ and $DD_{1,3}$. DD compensation (DDC) is typically performed at destination (centralized-DDC), using dedicated high speed buffers (reconstruction buffers) to store the sub-channels until recovering the original packet sequence. In centralized DDC, destination nodes must be equipped with large buffers (random access memories) operating at full-duplex speed (two times the line rate e.g., 2x40 Gbit/s or 2x100 Gbit/s) [237]. In Fig. 5.1 the destination node must compensate p_2 by 1ms and p_3 by 0.5ms to synchronize the MP connection. Assuming wavelength channels transmitting at 40 Gbit/s, the required total buffer size is 15 MB [2×40 Gbit/s $\times(1 + 0.5)$ms].

Centralized-DDC represents a bottleneck for MP which limits its performance gain by increasing: jitter in releasing data to the application layers, cost and complexity of the network nodes [238], not to speak of possible buffer overflows in case of DD exceeding the foreseen values.

This chapter presents the proposal of a novel optimization technique of MP routing in IP-over-WDM networks, that is based on fiber delay lines (FDL)s to introduce transparent DD compensation (transparent-DDC), that was presented in [239]. To the best of our knowledge, it is the first time that FDLs are proposed to perform DDC for MP routing. FDL-based transparent-DDC avoids the use of extra electrical-to-optical and optical-to-electrical (E/O-O/E) conversions of previous distributed electronic-DDC proposals [237, 240].

Commercially available compact FDL-modules, make our proposal a viable and interesting solution. For instance, the compact rack mount FDL offered by [241], that holds up to 90 km of fiber, can be integrated into an optical node to deploy transparent-DDC capabilities.

The following section 5.2 overviews related works on DDC. Section 5.3 describes the network architecture composed by nodes with FDLs and reconstruction buffers coupled with optical regenerators. Section 5.4 proposes a formulation for the MP routing problem with transparent-DDC and co-location

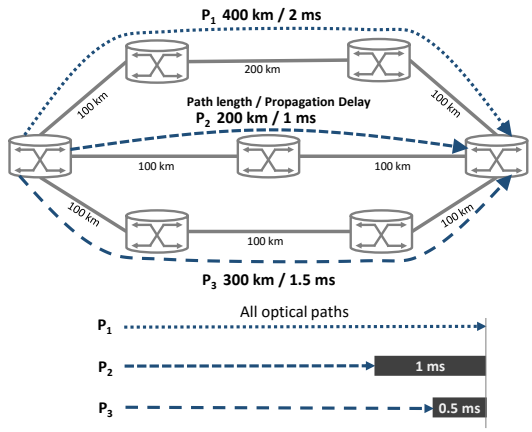

FIGURE 5.1: Common approach: centralized-DDC.

of optical regeneration and electronic-DDC. Section 5.5 reports numerical results obtained with realistic network topologies. Finally, section 5.6 concludes the work.

5.2 Related works

In circuit-switched optical networks there are two main approaches to mitigate the effects of DD in MP routing: 1) finding delay-similar paths by minimizing or constraining the *DD* of the MP set [27, 28, 29] (see Chapter 4), and 2) delaying the shorter paths of the MP set at intermediate nodes, as proposed for the first time by Alicherry *et al.* [30], hereafter called differential delay compensation (DDC) techniques. The first approach is constrained by the network topology, while the second allows more flexibility to mitigate the DD.

The common way of compensating the effects of differential delay in MP routing is to use reconstruction buffers at destination, hereafter called centralized-DDC. However, in presence of large DD the centralized-DDC technique requires very large reconstruction buffers and may lead to poor performance in terms of jitter and packet losses.

Alicherry *et al.* [30] proposed to distribute the reconstruction buffers along the paths with shorter delays to: 1) minimize the impact of DD at the destinations, and 2) reduce the reconstruction buffer requirements. In [30], distributed electronic-DDC was applied to VCAT in SONET/SDH networks. Later in [237], the distributed electronic-DDC problem was extended to optimize inverse multiplexing in OTN using a dual step ILP (integer linear programming) and two heuristic algorithms. Fig. 5.2 depicts how distributing the electronic-DDC along paths p_2 and p_3 allows to lower the maximum reconstruction buffer size from 15 MB, obtained with centralized-DDC (Fig. 5.1), to 5 MB [2×40

Gbit/s ×(0.5)ms)].

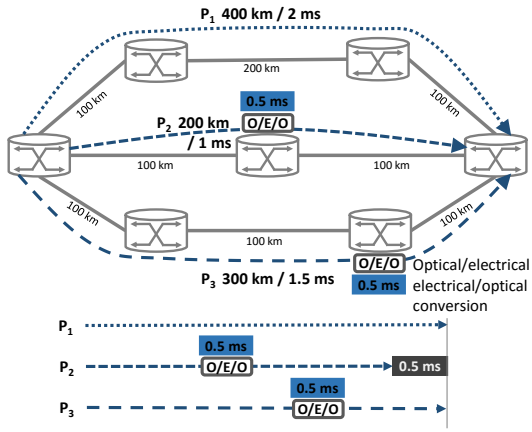

Figure 5.2: Distributed electronic-DDC.

However, in optical networks the distributed electronic-DDC incurs in extra costly and power-consuming O/E-E/O conversions to use reconstruction buffers along the paths. Such impact is clearly depicted in Fig. 5.2, where two extra O/E-E/O conversions are introduced into the MP connection to perform distributed electronic-DDC. While, in centralized-DDC (Fig. 5.1) the MP set is composed by three all optical circuits.

To lower the impact of distributed electronic-DDC in optical networks, Santos *et al.* [240] proposed to jointly place electronic-DDC and optical regeneration. Thus extra O/E-E/O conversions are avoided. Nonetheless, such approach greatly reduces the candidate nodes to distribute the DDC, as optical regeneration is only needed for very long lightpaths (longer than 1200 km). In fact, optical regeneration is only present in very large networks like NSFNET or European networks.

5.2.1 Contribution

This work proposes, for the first time, the use of compact FDL-modules to perform all optical distributed DDC (transparent-DDC) for MP routing. The FDLs are passive elements that overcome the problems of previous distributed electronic-DDC solutions for optical networks:

- Do not incur into extra O/E-E/O conversions. For instance, the power consumption of O/E-E/O conversion for 1 non-coherent channel at 40 Gbit/s is 200 W (power consumption of two transponders) [242].

- Are not restricted to optical regeneration points.

The power consumption of transparent-DDC is related to extra optical line amplifiers. For instance, the introduction of a FDL module of 100 km is 120 W [242], where up to 80 channels can be compensated. However, FDLs introduce a trade-off in flexibility when compared with electronic buffers, because each FDL produces a fixed delay.

Fig. 5.3 depicts the proposed transparent-DDC. DDC of paths p_2 and p_3 is performed by 100km FDLs. The FDLs allow the signals of p_1, p_2, p_3 to experience the same propagation delay. In the example, D is reduced to zero without including costly O/E-E/O converters or large high-speed buffers.

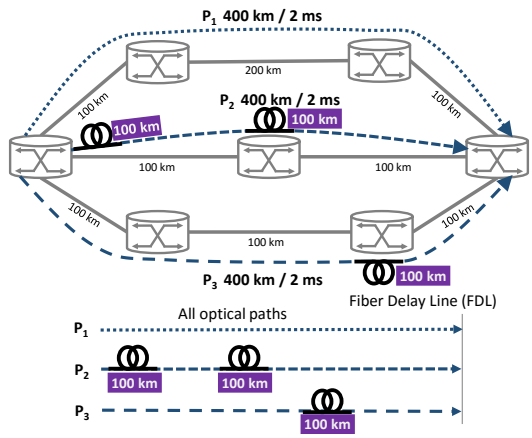

FIGURE 5.3: Proposed distributed transparent-DDC (with FDLs).

We proposed three new techniques, based on mixed integer linear programming (MILP) formulation, of MP link-disjoint routing assignment and hybrid DDC (transparent-DDC and electronic-DDC). These three techniques exploit transparent-DDC in combination with electronic-DDC: the difference between them is that electronic-DDC can be centralized, distributed or co-located with optical regenerators (see section 5.4). The following section describes the network scenario and the optical node (Fig. 5.4) with the proposed transparent-DDC capabilities. For brevity, we only illustrate the case when electronic-DDC is co-located with optical regeneration.

5.3 Network Scenario

This work focuses on WDM-based optical layer (lower layer), composed by Optical Cross Connects (OXC)s that provides connectivity via optical fiber links to an upper layer composed by IP core routers. Such architecture is known as IP-over-WDM (IPoWDM).

In IPoWDM, the demands are terminated (or dropped) and generated (or added) at the IP layer, like demands d_1 and d_2 in Fig. 5.4, respectively. In the

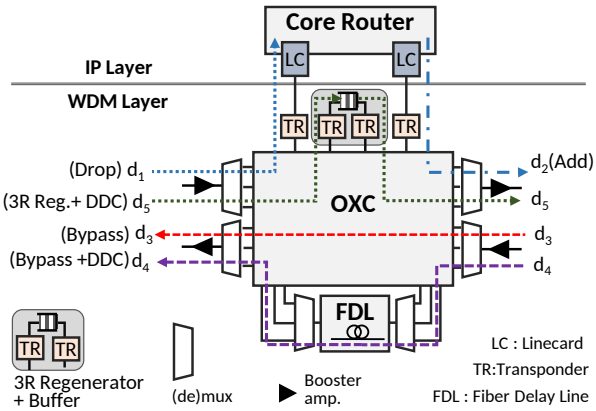

Figure 5.4: Optical node composed by: Optical cross connector (OXC), FDL module for Transparent-DDC, Electronic-DDC collocated with optical regeneration, and core router.

basic IPoWDM, all the wavelengths are terminated at each hop to perform traffic grooming and switching in the electrical domain. Thus, in every node all data flows undergo through O/E and E/O conversions.

Following the trend of reducing energy consumption in transport networks, this work considers transparent IPoWDM networks (T-IPoWDM). The T-IPoWDM is capable of routing optical circuits at the WDM layer, which are called lightpaths. In T-IPoWDM, the IP layer can be bypassed at intermediate nodes by configuring the optical switching matrices of the OXCs. The optical bypass allows to avoid power-hungry O/E-E/O conversions for transit traffic (e.g., demand d_3 in Fig. 5.4).

In the literature, distributed electronic-DDC is performed with costly high speed electronic buffers [30, 237]. As proposed in [240] for VCAT, in a large optical network the electronic-DDC can be co-located with optical regenerators.

Due to degradation of optical signals along the lightpath (e.g., cumulated noise, cross-talk and non-linear distortions), optical regeneration must be introduced for lightpaths longer than the regeneration span. Depending on optical-signal characteristics, such as the modulation format, the optical span can be 1200, 1500 or even 2000 km. The regeneration of optical signals is an electronic process comprising Reamplifying, Reshaping and Retiming (3R regeneration). It is performed at the optical layer with a pair of back-to-back placed transponders.

As depicted in Fig. 5.4, the O/E and E/O conversions done by the two back-to-back transponders of the 3R regenerator allow the introduction of a high speed buffer. Thus, if necessary electronic-DDC can be performed without incurring into extra O/E-E/O conversions. In Fig. 5.4, demand d_5 is switched by the OXC so to cross a 3R regenerator, inside of which it undergoes

Table 5.1: Evaluated DDC techniques

	DDC Technique	
MILP Model	**Electronic**	**Transparent**
C-Buff	Centralized	No
D-Buff	Distributed	No
FDL-CBuff	Centralized	Yes
FDL-DBuff	Distributed	Yes
FDL-RDBuff	Co-located 3R regenerator	Yes

electronic-DDC.

As described in section 5.2.1, the node shown in Fig. 5.4 is able to route lightpaths through a FDL module to introduce transparent-DDC, as for demand d_4. Demand d_4 experiences all-optical DDC of G seconds proportional to the physical length of the fiber coil selected for the compensation.

5.4 Optimization models

In this work we developed five MILP formulations for the DDC techniques presented in Table 5.1. However, due to space limitations, we only describe FDL-RDBuff model, which is the most complete one, and from which the others can be derived.

Consider an optical WDM network represented by a bi-directed graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ are the sets of nodes and directed links, respectively. Each demand $d \in \mathcal{D}$ is defined by a source and destination node pair, the required bandwidth volume in terms of number of required link disjoint wavelength channels h_d (e.g., $h_d = 4$ channels at 10 Gbit/s for a 40 Gbit/s request), and the differential delay upper bound Δ_d. The FDL-RDBuff problem consists in finding a MP routing solution that complies with the required bandwidth and D constraint Δ_d of all the demands $d \in \mathcal{D}$ using transparent-DDC and electronic-DDC co-located with 3R regenerators. To simplify the problem, we used a two step approach. The first step solves the FDL-RDBuff problem without wavelength continuity constraint. The second step solves the wavelength assignment problem using a graph colouring-based heuristic (the heuristic is not presented to simplify the presentation).

For each pair of nodes, a set of disjoint paths is pre-calculated using the k-edge-disjoint paths algorithm [243]. Any demand $d \in \mathcal{D}$ can be then associated to the precomputed set $\mathcal{P}_d$, described by the parameters δ_{edp} ($\delta_{edp} = 1$ if link e belongs to the path $p \in \mathcal{P}_d$ realizing demand d; 0, otherwise). The variables of the path formulation are the following:

- Binary Flow variables $x_{dp} = 1$ if path $p \in \mathcal{P}_d$ is used by demand d; 0, otherwise.

- Binary variables $R_{dpv} = 1$ if 3R reg. is placed in node $v \in \mathcal{V}$ for path $p \in \mathcal{P}_d$ realizing demand d; 0, otherwise.
- Non-negative variables b_{vdp} is the amount of electronic-DDC at node $v \in \mathcal{V}$ of path $p \in \mathcal{P}_d$ realizing demand d.
- Non-negative variables U_{dp} is the maximum electronic-DDC assigned to a node of path $p \in \mathcal{P}_d$ realizing demand d.
- Non-negative variables B_{dp} is total electronic-DDC of path $p \in \mathcal{P}_d$ realizing demand d.
- Non-negative variables T_{dp} is total transparent-DDC of path $p \in \mathcal{P}_d$ realizing demand d.
- Non-negative integer variables g_{edp} is the amount FDL-modules of length G for transparent-DDC at link $e \in \mathcal{E}$ of path $p \in \mathcal{P}_d$ realizing demand d.

DDC (amount of compensated differential delay) is always measured in ms.

FDL-RDBuff:

$$\min \sum_{d\in\mathcal{D}} \sum_{p\in\mathcal{P}_d} (\sigma_{dp} x_{dp} + T_{dp} + \alpha B_{dp} + \omega U_{dp}) \tag{5.1a}$$

$$\sum_{p\in\mathcal{P}_d} x_{dp} = h_d, \quad d \in \mathcal{D} \tag{5.1b}$$

$$\sum_{d\in\mathcal{D}} \sum_{p\in\mathcal{P}_d} \delta_{edp} x_{dp} \leq WF_e, \quad e \in \mathcal{E} \tag{5.1c}$$

$$x_{dp}\sigma_{dp} \leq \sigma_d^{max}, \quad d \in \mathcal{D}, p \in \mathcal{P}_d \tag{5.1d}$$

$$\sigma_d^{max} - x_{dp}\sigma_{dp} - T_{dp} - B_{dp} - \phi_d (1 - x_{dp}) \leq \Delta_d, \quad d \in \mathcal{D},\ p \in \mathcal{P}_d \tag{5.1e}$$

$$g_{edp} G \leq \psi, \quad d \in \mathcal{D}, p \in \mathcal{P}_d,\ e \in \mathcal{E} \tag{5.1f}$$

$$x_{dp} w_{sdp} + \sum_{e\in\mathcal{E}} \Pi_{sedp} g_{edp} \leq L(1 + \sum_{v\in\mathcal{V}} \Gamma_{svdp} R_{vdp}), \quad d \in \mathcal{D}, p \in \mathcal{P}_d, s \in \mathcal{S} \tag{5.1g}$$

$$b_{vdp} \leq \sum_{s\in\mathcal{S}} \phi_d \Gamma_{svdp} R_{vdp}, \quad d \in \mathcal{D},\ p \in \mathcal{P}_d,\ v \in \mathcal{V} \tag{5.1h}$$

$$b_{vdp} \leq U_{dp}, \quad d \in \mathcal{D}, p \in \mathcal{P}_d, v \in \mathcal{V} \tag{5.1i}$$

$$\sum_{v\in\mathcal{V}} b_{vdp} = B_{dp},\ d \in \mathcal{D}\ , p \in \mathcal{P}_d, \tag{5.1j}$$

$$\sum_{e\in\mathcal{E}} \delta_{edp} g_{edp} G = T_{dp}, \quad d \in \mathcal{D}\ , p \in \mathcal{P}_d, \tag{5.1k}$$

$$x_{dp}, R_{vdp} \in \text{binary},\ g_{edp} \in Z^+, \quad b_{vdp}, B_{dp}, g_{edp}, T_{dp} \in R^+ \tag{5.1l}$$

In the formulation, objective (5.1a) minimizes the overall delay of the demands $d \in \mathcal{D}$ as a linear combination of:

- Propagation delay ($\sigma_{dp} x_{dp}$).

- Transparent and electronic-DDC ($T_{dp}+\alpha B_{dp}$). $\alpha > 1$ represents the extra cost assigned to electronic-DDC to set the priority to transparent-DDC.
- Maximum electronic-DDC ωU_{dp} compensated in one path $p \in \mathcal{P}_d$, in order to distribute (if possible) the electronic-DDC ($0 < \omega < 1$).

The solenoidality and capacity constraints are formulated in (5.1b) and (5.1c). In (5.1c) the capacity of each link is given by the product between number of wavelengths W and fibers per link F_e. The maximum delay of the paths used by demand d (σ_d^{max}) is calculated in (5.1d). σ_{dp} is the delay of path p realizing demand d, while ϕ_d is the maximum delay of all candidate paths for demand d. The differential delay constraint is formulated in (5.1e). The maximum length of the FDL is upper bounded in (5.1f) by $\psi = 200$ km: this is a reasonable value requiring to install maximum 2 optical amplifiers (OA)s to compensate losses due to FDL (assuming, as usual, that one line OA is needed every 100 km of fiber [64]).

Constraint (5.1g) determines the candidate nodes v where to place optical regeneration ($R_{vdp} = 1$) for path $p \in \mathcal{P}_d$ realizing demand d based on:

- the regeneration span ($L = 1200km$),
- the distance traveled by path $p \in \mathcal{P}_d$ without optical regeneration (w_{sdp}),
- Π_{sedp} ($\Pi_{sedp} = 1$ if until hop s link e belongs to path p realizing demand d; 0, otherwise), and
- Γ_{svdp} ($\Gamma_{svdp} = 1$ if until hop s node v is visited by path p realizing demand d; 0, otherwise).

Based on R_{vdp} and Γ_{svdp}, the inequality (5.1h) enables co-location of electronic-DDC (b_{vdp}) in the candidate nodes for optical regeneration placement of path $p \in \mathcal{P}_d$ and demand d. Constraint (5.1i) determines U_{dp}. Equalities (5.1j) and (5.1k) calculate B_{dp} and T_{dp}, respectively.

5.5 Results

This section describes the results of the DDC techniques presented in Table 5.1 for two real networks:

- The NSFNET [244], with *Average_Link_Length* $\geq$ *Regeneration_Span* (1200 km), considered as a large network.
- The Polish network [245], with *Average_Link_Length* $<$ *Regeneration_Span*, considered as a small network

To solve te MILP problems we used ILOG CPLEX 12.0 on a workstation with 8 processors at 2.00 GHz and 32 GB of RAM. We first analyze the impact of the FDL-module length G, which determines the transparent-DDC granularity.

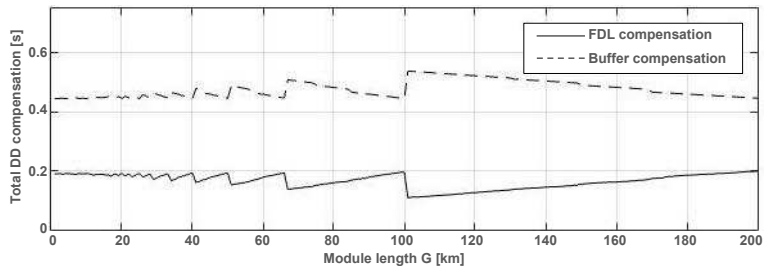

Figure 5.5: Transparent & electronic-DDC vs FDL Module length G, using FDL-RDBuff in NSFNET.

A trade-off exists: small G allows more flexibility but requires higher number of FDL modules, while large G limits the flexibility but reduces the number of modules. Figs. 5.5 and 5.6 depict the total amount of transparent-DDC and electronic-DDC installed in the NSFNET and Polish networks by solving a hybrid DDC for $G \in \{1, 2, 3, ..., 200\}$ km. More specifically:

- FDL-RDBuff (described in section 5.4) was used in a large network (NSFNET), that requires 3R regeneration.
- FDL-DBuff was used in the smaller network (Polish), where 3R regeneration is not required.

In both scenarios, 50% of all the possible pair of demands in the network where active, and the most stringent differential delay upper bound $\Delta_d = 0$ was imposed.

We can notice that in the large network (Fig. 5.5), the total transparent-DDC is smaller than the electronic-DDC. Alternatively, in the small network (Fig. 5.6), the total transparent-DDC is larger than the DDC realized in the buffers. In both network scenarios, the electronic compensation reached local minima for FDL-module lengths $G \leq 10 \cup \{20, 25, 33, 40, 50, 66, 100, 200\}$ km, hereafter G^{min}. It should be noticed that for $G \in G^{min}$, the transparent-DDC ($g_{edp}G$) allocated for a path $p \in \mathcal{P}_d$ can be equal or almost equal to the maximum allowed transparent-DDC $\psi = 200$, imposed by inequality (5.1f). While, for $G \notin G^{min}$, the gap between $\psi = 200$ and $g_{edp}G$ is larger.

An important insight is that the minimum values of electronic-DDC in Figs. 5.5 and 5.6, remain almost the same for small and large values of $G \in G^{min}$. Therefore, the discrete nature of the FDL delay does not impose a big penalty in the performance of transparent-DDC techniques (even for the most stringent D requirement: $\Delta_d = 0$).

Fig. 5.7 depicts the total buffer size installed in the network for electronic-DDC when solving the hybrid DDC models of table 5.1, in NSFNET and Polish networks. The buffer size is measured in terms of compensated delay [s]. Results are obtained under three load scenarios: $\{10, 30, 50\}$% of all possible demands

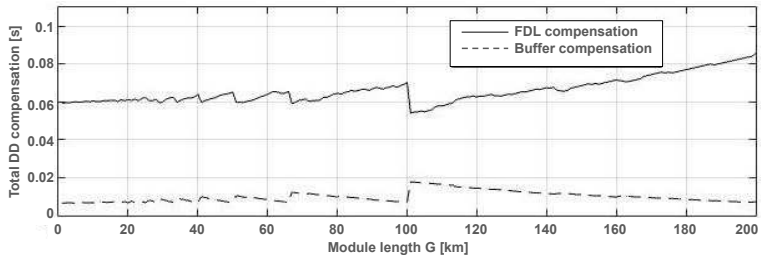

FIGURE 5.6: Transparent & electronic-DDC vs FDL Module length G, using FDL-DBuff in Polish Network.

in the network are active. Fig. 5.7 shows for both networks a clear reduction of electronic buffer utilization thanks transparent-DDC. In Polish (small) network, the models that use transparent-DDC (FDL-CBuff and FDL-DBuff) reduce the total buffer size by up to 80% when compared with models that only use electronic-DDC (C-Buff and D-Buff). In NSFNET (large) network, the models that use transparent-DDC (FDL-CBuff and FDL-RDBuff) reduce the total buffer size by up to 55% when compared with the models that only use electronic-DDC (C-Buff and D-Buff). Such gain difference can be also appreciated when comparing Figs. 5.5 (larger network with larger DD) and 5.6 (small network with smaller DD). However, in this regard it should be notice that the transparent-DDC $g_{edp}G$ was limited to $\psi = 200$km per path.

In a large network, the FDL-RDBuff model allows to jointly perform optical regeneration and electronic-DDC avoiding extra O/E-E/O conversions to use distributed electronic-DDC. In the case of small networks where the optical regeneration is not necessary, there are two options to implement the electronic-DDC: distributed and centralized. The former relaxes the requirements on buffer size but increases extra O/E-E/O conversions (FDL-DBuff); the later implies no extra O/E-E/O conversions but higher buffer size requirements (FDL-CBuff). Under the described performance evaluation scenario, FDL-CBuff performs as good as FDL-DBuff but with a reduction up to 88% of total electronic-DDC when compared to C-Buff. By properly designing the placement of the proposed transparent-DDC in small networks, the requirements on both distributed and centralized DDC are greatly reduced. This behavior suggests that FDL-CBuff with totally transparent distributed DDC and small centralized DDC is more convenient than the FDL-DBuff and FDL-RDBuff for small networks.

Fig. 5.8 depicts the size of the FDLs installed in the two networks under different traffic loads. For both networks, the total size of FDLs remains almost independent from the electronic-DDC technique (centralized or distributed). Clearly, the total FDL size required for the Polish network is smaller than the NSFNET, due to network dimensions. However, the difference among total FDL size experienced by both networks is smaller than the difference in total electronic buffer presented in Fig.5.7, because for both networks the same upper

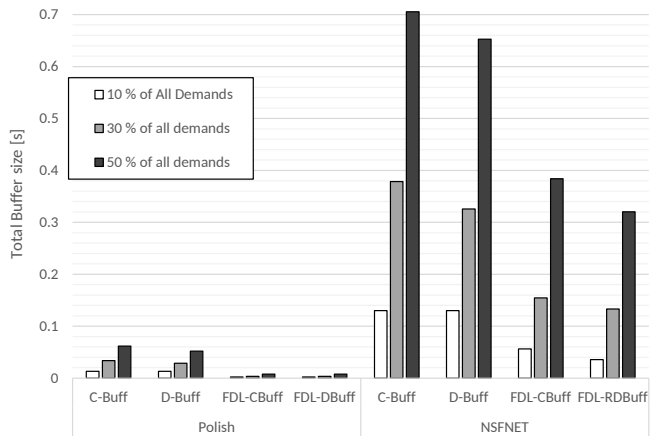

Figure 5.7: Total Buffer size of Electronic-DDC when solving CBuff, DBuff, FDL-CBuff, FDL-DBuff and FDL-RDBuff for Polish and NSFNET networks.

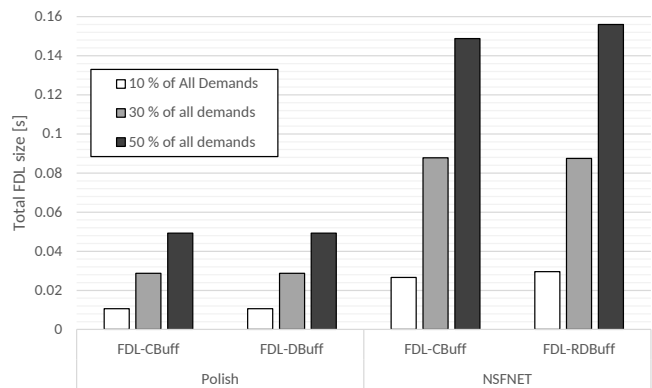

Figure 5.8: Total FDL size of transparent-DDC when solving FDL-CBuff, FDL-DBuff and FDL-RDBuff for Polish and NSFNET networks.

bound ($\psi = 200$) was set to the transparent-DDC per path.

5.6 Concluding Remarks

The novel FDL-based transparent-DDC technique proposed in this work is an interesting solution to reduce the impact of DD in MP routing over optical networks, without the limitations of electronic reconstruction buffers. Transparent-DDC has lower operating expenses than electronic-DDC. Although FDLs introduce fixed delays, our results showed that the discrete nature of FDLs does not impact the performance of the hybrid DDC models (transparent-DDC and electronic-DDC). The hybrid DDC techniques were proven to be

better than C-Buff and D-Buff. Transparent-DDC avoids extra O/E-E/O conversions and reduces the size of reconstruction buffers. The O/E-E/O conversions can be further avoided by properly choosing the electronic-DDC approach: distributed electronic-DDC co-located with 3R regeneration for large networks, and centralized electronic-DDC for smaller networks.

Commercially available rack mount FDL-modules [241], make our proposal very interesting for real implementation of transparent-DDC capabilities in optical nodes.

Future works should elaborate on enhanced optimization models for the design of cost-effective optical networks using transparent-DDC and electronic-DDC.

Demonstration of SDN-Based Orchestration for Multi-Domain Segment Routing Networks 6

This Chapter demonstrates a hierarchical control plane architecture for Software Defined Networking (SDN)-based Segment Routing (SR) in multi-domain networks. An orchestrator application, on top of multiple open source SDN controllers, creates a hierarchical control plane architecture using northbound RESTFul APIs of controllers. The orchestrator has control, visibility and traffic engineering capabilities to manage multi-domain SR service creation. Standard southbound interfaces with proper SR extensions are exploited to manage SR tunnels in the MPLS data plane.

6.1 Introduction

Segment Routing (SR) is a new source routing paradigm (announced by Cisco in 2013) intended to provide traffic engineering (TE) solutions, and to address several control plane drawbacks of legacy IP/Multiprotocol Label Switching (IP/MPLS) networks in terms of scalability, simplicity, and ease of operation. SR currently under standardization by the IETF Source Packet Routing in Networking (SPRING) working group [246].

In SR, segment identifiers (SID)s are labels that represent intermediate path points, and are encoded in 32 bits MPLS labels. Paths are specified at the source node as an ordered list of SIDs, which is equivalent to MPLS label stack. SR can be leveraged in existing MPLS networks with a small set of extensions to routing protocols such as: Interior Gateway Protocols (IGPs) and Border Gateway Protocol (BGP) and Path Computation Element Protocol (PCEP) [247, 248].

Routers use the forwarding information base (FIB) to forward packets to

the shortest path towards the SID network elements. Thus no longest-prefix matching is needed and routers switch packets based on labels. With SR, the number of entries in a router FIB is generally proportional to the number of nodes in the networks and the number of links connected to the router, and is thus static. On the contrary, with MPLS, the number of states stored in a forwarding table depends on the number of active label-switched paths, which can grow substantially and is a dynamic parameter. Moreover, by implementing SR the network avoids the scalability issues of Resource ReserVation Protocol-TE (RSVP-TE), and the heavy signaling burden of Label Distribution Protocol (LDP), making network operation simpler.

SR leverages on the presence of a centralized element that calculates paths using a global view of the network, and sends the path to the source node of the traffic flow as an ordered list of SIDs. Thus, SR can be used to enable Software Defined Networking (SDN) and application engineered routing in carrier networks. Carrier networks are normally composed by heterogeneous multi-domains, with autonomous systems parted in subnets implemented by multiple vendors and with different technologies.

SDN standardization bodies agreed upon hierarchical control plane architectures for the carrier networks [249]. Interoperability among heterogeneous control planes can be better tackled by an orchestration application on top of the control plane. The orchestrator, using open standard RESTFul interfaces, interacts with abstracted view of the heterogeneous domains for end-to-end service coordination [46].

In this work we demonstrate the orchestration of a multi-domain SDN-based Segment Routing network. A hierarchical control plane architecture is created by an Orchestration application that uses the northbound interface of multiple SDN controllers. The hierarchical architecture is compatible with recommendations from ONF [249], IETF southbound protocols [246, 247] and northbound APIs from open source SDN controllers. The Orchestrator allows cross-domain SR capabilities over SDN controlled heterogeneous domains.

6.2 Related works

As technology and as a standard, SR is at very early stages of development, and it is starting to gain attention in the scientific community. For instance, SR scalability issues are studied in [250] and [251]. The work presented in [250] proposed two routing techniques to improve the scalability of SR, while the research in [251] implemented SR algorithms for flow assignment that try to minimize overall network crossing time, thus reducing the size of the SID list for packets in the network.

SR is compatible with a centralized control plane and SDN, and it is an interesting option to combine the advantages of distributed control planes (e.g., MPLS and GMPLS) and centralized control planes (PCE and SDN) over different network scenarios. [139] demonstrated the first SDN-based SR imple-

mentation for multi-layer packet-optical network. A Ryu-based SDN controller (SR-controller) was extended to control the label stacking configuration at source nodes (routers, implemented as Open vSwitches). The SR-controller utilized OpenFlow 1.3 to interface Open vSwitches, while commercial ROADMs with 10Gb/s OTN muxponders, that were used to provide an optical bypass.

Later, SR was implemented in IP/MPLS networks using a path-computation-element (PCE) architecture [252]. Extensions to the PCE protocol (PCEP) allowed a centralized PCE to control the label stacking configuration. Both SR implementations using SDN [139] and PCE [252] were able to perform dynamic packet rerouting (with optical bypass capabilities in [139]), by enforcing different segment-lists at source node, without any signaling protocol and with no packet loss.

In [140, 139] was extended for the multi-domain network scenario, using a mesh of domain-specific SR-controllers. The SIDs advertisement is limited to each domain, thus SR-controllers do not have access to other controller's intra-domain topology. Due to the mesh control-plane architecture, there is no global view of the network, thus controllers do not know the domain sequence to reach destination. However, [140] assumed that such sequence of domains is known. The paper proposed two methods for SID-list exchange among the domains using a non-standard east/west interface between peer controllers, thus relying on a flat control-plane architecture.

On the other hand, to foster scalability and interoperability in multi-domain networks, standardization efforts in SDN and PCE recommend hierarchical control plane architectures, that adopt standard north bound interfaces to provide end-to-end service orchestration [249, 46]. SR suits well with a hierarchical control-plane because it has been developed for centralized control architectures.

6.2.1 Contribution

To our knowledge, there is still lack of works on: 1) hierarchical control-plane architectures implementing SR and 2) standard methods to exchange SID information in a multi-domain scenario. This gap justifies the implementation and experimental activity we have carried out and we are reporting in this Chapter, together with some preliminary results achieved with a standard-based hierarchical SDN control-plane, created by an Orchestration application on top of SR-controllers. The hierarchical approach becomes realistic in a single-carrier multi-domain network, or when different carriers agree upon the adoption of a common coordination box. That does not include all possible situations, but still applies in relevant cases.

6.3 Hierarchical Architecture

Fig. 6.1 shows the architecture of our multi-domain hierarchical SDN-based Segment Routing network. The data-plane layer is at the moment represented

by a multi-domain MPLS network (which will be extended to include other layers, such as the optical layer, in future developments). In each domain, an SDN controller provides abstraction and control of data plane. We use standard southbound interfaces; BGP-LS and PCEP of SDN controller to get network topology and to push MPLS SR tunnels. One router in each domain runs BGL-LS speaker and advertises the network topology to the controller, while every router in the network runs Path Computation Client (PCC) to enable tunnel creation and modification using PCEP.

On top of the SDN controllers, a higher-level Orchestrator application creates a hierarchical control plane architecture for multi-domain networks. Using the northbound interfaces of the SDN controllers (RESTFul APIs), the orchestrator provides complete control, visibility and TE capabilities for cross-domain traffic, using segment routing. Moreover, Orchestrator software application abstracts the multi-domain network and provides its own RESTFul APIs to enable intelligent applications and traffic engineering solutions for the network.

Our architecture is flexible and can be adopted in a multi-vendor environment. Indeed, the orchestrator gets abstracted information from the controllers and thus can in principle be connected to any vendor-provided domain controller. Thus, proprietary vendor data-plane equipment can be controlled, even without knowing what southbound protocol it uses to connect to the controller.

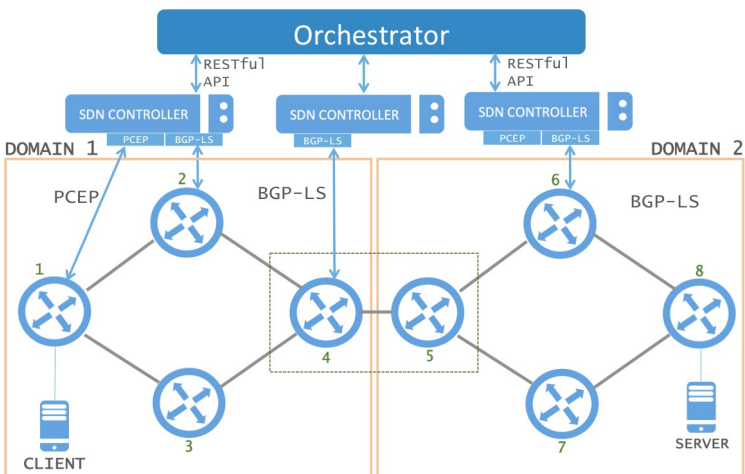

Figure 6.1: General multi-domain hierarchical SDN Architecture.

6.4 Implementation and use cases

The data plane was implemented using Cisco IOS XRv routers running inside VirtualBox. XRv emulates a real router and supports all routing protocols,

PCEP and MPLS data plane. Configuration steps for routers include: 1) configuring proper IP addresses to all interfaces; 2) assigning SIDs to node and interfaces; 3) configuring MPLS data plane, and 4) configuring PCEP. Step 2) requires instantiating Intermediate System-to-Intermediate System (ISIS) protocol in each domain to exchange route information and SIDs between routers. By default, routers do not send MPLS traffic from interfaces: thus step 3) is needed to add each interface in MPLS configuration. By step 4) we tell each router to use PCC based auto-tunnels.

We used OpenDayLight SDN controller (Lithium release) in each domain and no changes have been made to controller. Each controller is configured to speak with a BGP-LS router in each domain (the BGP-LS speaker). Orchestrator-application software reads topology information from each controller using RESTFul APIs and creates a global multi-domain network topology. Because of limitations of ODL and xRV, it is challenging to identify gateway nodes and inter-domain connectivity information in network. We used an additional controller for border nodes so Orchestrator is able to precisely gather the complete inter-domain information. Orchestrator software includes a path computation engine to find resource-constrained paths on the network and it uses RESTFul PCEP APIs (provided by SDN controller) to push tunnel-setup messages to the source router.

Routers build MPLS Forwarding Information Base (FIB) by exchanging SID information via IGP protocol. Fig. 6.2 shows the FIB of router 1 corresponding to implementation of the network depicted in Fig. 6.3. Paths for the traffic in the network can be engineered by using appropriate stack of prefix SID or adjacent SID. When a router receives a packet, it consults its FIB and routes the packet on appropriate interface towards the next hop. For prefix SIDs, routers forward the packet using shortest path, provided by IGP, towards the node indicated by SID. For adjacent SIDs, routers forward the packet directly on the interface indicated by SID.

6.4.1 Equal Cost Multipath (ECMP) use case

Using ECMP it is possible to distribute the traffic among multiple disjoint equal-cost paths. When routers build FIB, if there are multiple equal-cost paths, they associate all those paths to a prefix SID. For instance, in 6.2, router 1 has two equal paths to reach router 4 and hence the FIB for label 4 has two entries (one for each path). ECMP leads to simpler FIB and reduces the size of label stack inside each packet.

6.4.2 Application-aware SR use case

Different applications may have different resource requirements, e.g., bandwidth and delay constraints. To perform application-aware SR, we encoded the path using adjacent SIDs. Orchestrator application finds suitable paths, composed of links that meet the requirements for an application, and pushes it to the

MPLS Forwarding Table for router 1

```
RP/0/0/CPU0:n1#show mpls forwarding
Fri Apr 29 11:20:05.874 UTC
Local  Outgoing    Prefix             Outgoing     Next Hop
Label  Label       or ID              Interface
------ ----------- ------------------ ------------ ----------
2      Pop         No ID              Gi0/0/0/0    10.10.12.2
3      Pop         No ID              Gi0/0/0/1    10.10.13.3
4      4           No ID              Gi0/0/0/0    10.10.12.2
       4           No ID              Gi0/0/0/1    10.10.13.3
12     Pop         SR Adj (idx 1)     Gi0/0/0/0    10.10.12.2
13     Pop         SR Adj (idx 1)     Gi0/0/0/1    10.10.13.3
```

MPLS Forwarding Table for router 8

```
RP/0/0/CPU0:n8#show mpls forwarding
Wed Apr 13 14:02:08.039 UTC
Local  Outgoing    Prefix             Outgoing     Next Hop
Label  Label       or ID              Interface
------ ----------- ------------------ ------------ ----------
5      5           No ID              Gi0/0/0/0    10.20.68.6
       5           No ID              Gi0/0/0/1    10.20.78.7
6      Pop         No ID              Gi0/0/0/0    10.20.68.6
7      Pop         No ID              Gi0/0/0/1    10.20.78.7
86     Pop         SR Adj (idx 1)     Gi0/0/0/0    10.20.68.6
87     Pop         SR Adj (idx 1)     Gi0/0/0/1    10.20.78.7
```

Figure 6.2: Forwarding Information Base (FIB) generated by router 1 in Domain 1 of the general architecture.

source router. SR using adjacent SID gives complete control and flexibility to engineer the exact path. Orchestrator creates different tunnels for different applications. In the current implementation, we were obliged to use static route configuration on each router to transfer application traffic via the tunnel that is created by the controllers: this limitation descends for the version of XRv we adopted (5.3.2) that does not support automatic tunnel creation. With an improved virtual router, it will be easy to shift to a fully-automatic mechanism in the future.

6.4.3 Multi-domain use case

For traffic that crosses multiple domains, the orchestrator must find the end-to-end path depending on the routing policies and resource availability of each domain. Thanks to the use of hierarchical architecture, the Orchestrator application gets an abstracted view of the unified topology and resource availability of the entire network. Note that the overall view is based only on SIDs and each domain can decide what SIDs to export: that guarantees confidentiality of detailed intra-domain topology information, making our approach potentially suitable also for a multi-carrier scenario. Orchestrator reads the topology from all domain controllers and then runs graph-based algorithms to identify border routers and intra-domain connectivity. Orchestrator can run advanced algorithms to find best paths for traffic using prefix SID or adjacent SID that not only increase network utilization but also meet application requirements.

Our hierarchical architecture is highly scalable because SDN controllers do not exchange any SID information among each other. Routers only exchange SID information with other routers in the same domain and as shown in Fig. 6.2, FIB of router 1 does not include any SIDs of domain 2. Moreover, our design is simple and compatible with standards and recommendations from ONF [249] and IETF [246, 247], as it is based on standard southbound protocols (BGP-LS and PCEP), and RESTFul APIs of an open-source SDN controller (ODL).

6.5 Results

Fig. 6.3 presents an example of multi-domain TE using SR. Domain 1 has ECMP policies to maximize network utilization, while domain 2 have strict policies to enable better QoS. There are two servers, each running two different applications. Orchestrator application, based on the input policies, finds suitable SID stack and via PCEP sends that SID path to source router, which encodes the path in each packet from client. Orchestrator encodes the path as an Explicit Route Object (ERO), then using northbound PCEP tunnel-create message, it sends the path to the domain controller of source node.

Fig. 6.4 shows an example of ERO with prefix SID and adjacent SID. SDN controller uses PCEP protocol messages to push the path to source router creating a Label Switched Path (LSP) tunnel from source to destination. Fig. 6.3 depicts the SID stack and traffic path for two applications. Application 1 has SID stack $\{4, 45, 56, 68\}$ and it takes shortest path from router 1 to router 4, then it follows a strict path crossing router 5, 6 and router 8. Application 2 has SID stack $\{4, 45, 57, 78\}$ and it also takes shortest path from router 1 to router 4 but in domain 2 it takes a strict path crossing router 5, 7 and router 8. Domain 1 does ECMP for the traffic and evenly distributes the traffic on both paths.

In our implementation, we do per-destination ECMP. Traffic for server 1 takes path 1-2-4 and traffic for server 2 takes path 1-3-4. Using static routing or policy-based routing, we can map client's traffic to different tunnels that Orchestrator creates. Table 6.1 shows the paths that client's traffic will take depending on the server and application type.

Table 6.1: Traffic paths of example in Fig. 6.3.

Server	App	Path
1	1	1,2,4,5,6,8
1	2	1,2,4,5,7,8
2	1	1,3,4,5,6,8
2	2	1,3,4,5,7,8

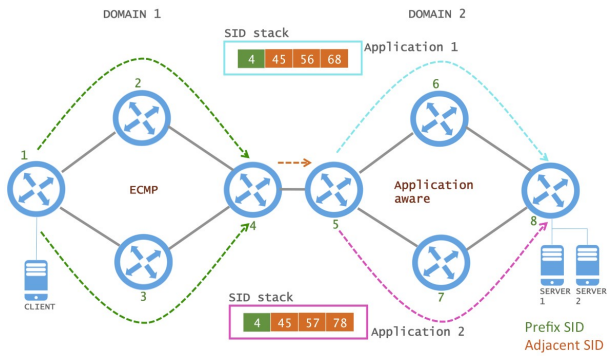

Figure 6.3: Multi-domain TE use case.

Explicit Route Object (ERO) encoded in PCEP

```
<ero>
        <subobject>
                <loose>false</loose>
                <sid-type>ipv4-node-id</sid-type>
                <m-flag>true</m-flag>
                <sid>4</sid>
        </subobject>
        <subobject>
                <loose>false</loose>
                <sid-type>ipv4-adjacency</sid-type>
                <m-flag>true</m-flag>
                <sid>45</sid>
        </subobject>
</ero>
```

Figure 6.4: Part of an Explicit Route Object (ERO) example.

6.6 Concluding Remarks

Multi-layer hierarchical Segment Routing implementation provides scalable, flexible and simpler architecture with complete control of path creation and path policies from a centralized Orchestrator. This increases network utilization and efficiency by incorporating ECMP in a dense network and also better QoS for applications by using constraint based path selection. It enables network providers to add new value to their business as they can dynamically address new requirements from customers. Moreover, the proposed architecture provides a standard method to exchange SID information in a multi-domain scenario.

Our centralized Orchestration layer provides a platform for smart business applications and traffic engineering solutions by abstracting the underlying multi-domain topology and giving a unified network view to higher level applications using RESTFul APIs.

This work represent a starting point for the development of applications that can further optimize and automate the service provider network. SR solution by

reducing the size of SID stack as in [253], or provide scheduled traffic engineering use cases like bandwidth calendaring. With SR, only the edge routers need to be controlled by the SDN control plane, thus it represents a interesting option to enable T-SDN without replacing the whole transport network.

SDN based Automated Testbed for Evaluating Multipath TCP 7

Multipath TCP is an extension to TCP protocol that allows a single connection to be split across multiple paths. MPTCP overcomes few limitations of TCP thereby offering added benefits like increased throughput and network resource utilization and resilience to possible link failures. Different data paths over the network face different end-to-end path delay. The performance of MPTCP is affected not only by the delay but also by the difference in delays experienced by various paths of the same connection. MPTCP runs on the host, and it is not aware of the underlying network topology. Hosts cannot create optimal paths to destination without assistance from a network control element. We use SDN technology to provide this assistance to MPTCP and to setup best available paths for MPTCP connections. In this Chapter we present the design of an SDN-based path manager for MPTCP, the implementation of an automated testbed for performance evaluation, and an in-depth analysis of different scheduling algorithms to understand how delay and differential delays between the paths affect the overall performance of MPTCP protocol.

7.1 Introduction

A vast majority of applications use Transmission Control Protocol (TCP) to transport their data reliably across the Internet. TCP protocol works at transport layer on top of Internet Protocol (IP), which operates at the network layer. IP provides unreliable datagram service and ensures that any host can communicate with any other hosts on the network. Network links could fail, and the decoupling of TCP from IP allows the network to reroute packets around failures without affecting TCP connections. Multipath TCP (MPTCP)

is a major extension to TCP, allowing the use of multiple paths between two end-hosts for the transmission of a single data stream [232].

MPTCP was originally conceived primarily to support multihoming [254], i.e. to allow a host having multiple connections to the network.

MPTCP can be used to improve network performance. For instance, data centers, which dominate computing landscape today, have thousands of servers running distributed applications to spread computation and storage resources across multiple servers. Data-center core networks can suffer congestion of bottleneck links. MPTCP can ease these situations by enabling load distribution and balancing, leading to an efficient use of parallel paths for different data center topologies. MPTCP has been proposed as a replacement for TCP in data centers as it can seamlessly exploit available bandwidth giving improved throughput and better fairness [255].

MPTCP can significantly improve performance and resiliency of current networks for different kinds of scenarios, like data centers, enterprise networks, carriers and mobile hosts. Layer-2 approaches and application-layer techniques to support multihoming do not achieve optimal benefits [256]. Conversely, a transport layer solution like MPTCP is simple to implement and effective.

TCP is a complex protocol, and MPTCP extension increases the complexity even further. Various factors affect the performance of MPTCP connections [257, 258], like scheduling algorithms, data-buffer size and path diversity vs. path jointness. A comprehensive evaluation of different factors can give us an insight into performance of MPTCP connections and will help us in finding better ways to fine-tune the parameters to get maximum benefit out of MPTCP. In particular, in this work we will focus on route-related factors, showing how an SDN-based path manager can act on them to augment MPTCP performance.

In MPTCP, a host opens multiple subflows for the same connection and strips data across subflows. Pooling of sub flow's resources is achieved by multiplexing the data segments across different subflows. Delay difference between multiple subflows can reduce the throughput of the overall connection. When two hosts communicate over a packet network, the path taken by a packet travelling across the network depends on a large number of conditions like network topology, routing protocols, routing policies, etc. Different paths exhibit different characteristics of bandwidth, congestion, and delay. The default path over the Internet may not be the best for a particular source and destination pair. There may be large numbers of alternative paths with better features. Work on Internet measurements has shown that redundant paths are common between pairs of hosts and that one can often achieve better end-to-end performance by adaptively choosing an alternate path [259].

We can control the number of subflows created for each connection and the association of subflows to different paths. There are two ways: (i) uncoordinated control, when the end-host decide on their own and subflows take the default path; (ii) coordinated control, when a controlling element acts as a path manager for all the hosts in the network. It has been shown in [260] that uncoordinated approach is simpler to implement but it can suffer from poor performance.

Coordinated control is far better performing and is intrinsically fairer. What this implies is that path selection is a significant factor in deciding the performance gains of MPTCP over TCP. In particular, performance gain largely depends on whether the subpaths are disjoint or not in the network. This Chapter's focus is primarily on exploiting path diversity and evaluating the performance based on routing characteristics of multiple subflows. To route different subflows over different paths, we need the help of the routing infrastructure. Differently from other previous proposals, relying on distributed protocols (e.g. LIP [254]), we use an Openflow-based SDN controller to implement a MPTCP multipath engine, configure switches to forward different subflows over different paths and study the effect of different characteristics on MPTCP performance.

Typically, an SDN controller can create flows from one host to another host based on the given network topology, but, the flow that a controller creates for general traffic like TCP or UDP in most cases follows the default path that is available from source to destination. Some SDN controllers do additional processing for flow creation and balance the load across multiple switches. However, these functionalities are usually pre-configured and pre-provisioned on the controller and are not well suited for MPTCP connection, where flows should be dynamically created for each connection based on the current state of the network. As we will show next, not only path disjointness is important, but also the features of the chosen paths have great impact on MPTCP performance, and in particular the end-to-end delay and the available bandwidth of each subflow.

To evaluate the performance of MPTCP, we must test the protocol with a broad range of values of delays and bandwidth on each path. A real network may not present the complete range of parameter-values to perform these test-cases. Moreover, parameters may fluctuate a lot over the duration of an experiment in progress. A more suitable approach is to do measurements on a simulated controlled network and then use a real network to validate the findings of the controlled network. We implemented an automated testbed that, based on a set of input parameters, configures the simulated network and performs data transfer from one host to another to measure the performance of MPTCP. With an automated testbed, we can do measurements over complete ranges of parameter variations and with very fine granularity.

Manual evaluation would be very time-consuming because firstly we must reconfigure the network each time as per the given test parameters and then we must record and analyze the results of each test case. Our automated testbed speeds up the process by taking hundreds of test cases as input, executing each test case automatically without any interruption and by evaluating the results using tracing tools.

The chapter's structure is as follows: Section 7.2 discusses the related work on MPTCP performance-evaluation with SDN technology. Section 7.3 describes the design of SDN-controlled MPTCP network and path manager. We discuss the implementation of automated testbed in section 7.4. We present the results of our work in section 7.5. Finally, we conclude our work and discuss future

development in section 7.6.

7.2 Related works

MPTCP scheduler is responsible for multiplexing the data over multiple subflows and the choice of scheduler for a given network and application can have a significant impact on the performance of MPTCP. Poor scheduling decision can cause head-of-line (HOL) blocking, a phenomenon where packets from lower-delay path wait for the packet from higher delay path. HOL blocking causes burstiness in the data stream and increases the undesirable jitter causing high response time and poor user experience for real-time applications [261]. A crucial parameter affecting the overall throughput of the MPTCP connections is the size of receive buffer necessary to reorder packets in the event of out of order data reception or packet loss. MPTCP receiver must provide sufficient buffer space, depending on the characteristics of different paths, to utilize completely the available capacity on different subflows [258]. Techniques like Retransmission and Penalization (RP) and Bufferbloat Mitigation (BM) have been suggested in the literature [257, 262] to mitigate the effect of poor scheduling decision.

Linux kernel MPTCP implementation supports two types of schedulers: Least-RTT-first and Round Robin. Paasch et al. [263] studied the performance of MPTCP schedulers for bulk data transfers and application-limited flows, presenting goodput and application-delay results for different schedulers and mitigation techniques.

Numerous experimental studies have recently been published that discuss performance and evaluation of current MPTCP implementations [264, 265], resulting in improved MPTCP congestion control algorithm. Latency measurements for MPTCP over parallel 3G and WiFi networks demonstrate that MPTCP proves to be a robust data-transport mechanism [266] and that can facilitate smooth handover for mobile/wireless [267]. Work done on multipath communication architectures for cloud access and inter-cloud communications using LISP and TRILL protocol show a need for a global view to enforce path-disjointness, to control computational-load balance and improve performance of the MPTCP [254]. An SDN-based framework for MPTCP performance evaluation on large-scale public network is presented by Sonkoly et al [268]. Their work focuses more on the control and measurement framework for large scale SDN network and presents the results obtained on these network using MPTCP. Our work is focused more on the design and implementation of local simulated SDN based MPTCP test-bed that can evaluate MPTCP protocol performance using different parameters. As pointed out before, a real network may not present the complete range and consistent parameter-values to perform evaluation. We believe that our approach of using a fast, controlled and simulated automated test-bed to evaluate the performance of MPTCP is much more efficient and the results obtained using our test-bed can then be validated

using large scale real networks.

In this Chapter we propose a unique approach of using SDN based MPTCP path manager that can reactively create optimum paths for MPTCP, an automated test-bed implementation to quickly evaluate MPTCP protocol and we present the results we obtained using our SDN framework and automated test-bed on the effect of differential delay on MPTCP throughput.

7.3 SDN hierarchical architecture

The architecture of SDN-based path manager for MPTCP is represented in Fig. 7.1. The system consists of three layers. Bottom layer is the data plane layer that can either be a real network of switches or a virtualized network infrastructure that runs the automated testbed application. Network layer is managed by an SDN controller. Our SDN controller is based on the open source framework OpenDayLight (ODL) [31]. The SDN controller parses flow requests from network layer to detect MPTCP flow requests. If one is detected, it forwards the request to an MPTCP path manager (MPM) network application running on top of ODL controller. MPM NetApp runs complex algorithms to find suitable paths for MPTCP connections.

Let us now describe the details of our solution. In SDN, the northbound interface provides network abstraction and control logic interface to the NetApps running on top of the controller. The Application Programming Interfaces (APIs) exposed by the controller northbound are a powerful and simple mechanism to make all the essential functionalities of the underlying data plane elements accessible to the NetApps and to support smart pro-active and reactive flow programming.

ODL currently offers two kinds of northbound interfaces: RESTful and Open Service Gateway initiative (OSGi) [31]. RESTful interfaces use XML/JSON over HTTP protocol, technologies that most programmers are familiar with and that many web services already use. For this reason, they are simple to use. Several well-supported libraries are available to the NetApp programmer for different programming platforms. Parsing and creating JSON or XML messages as well as sending and receiving them over HTTP are straightforward. However, because of the request/response nature of REST and HTML, these interfaces are restricted to proactive flow programming.

OSGi control applications and services can use any feature exposed by the controller or by other OSGi bundles. OSGi is much more complex than RESTful interfaces but more powerful, as it allows re-active flow programming. It should be noted that, on the other hand, OSGi is less versatile, as it is designed for Java applications and cannot be used to for any other programming language. Comparing RESTful to OSGi, there seems to be a trade-off between simplicity (of RESTful) and flow-reactivity (of OSGi).

To dynamically manage routing of MPTCP connections in the MPM we needed both the complex graph-based algorithms available in Python-language

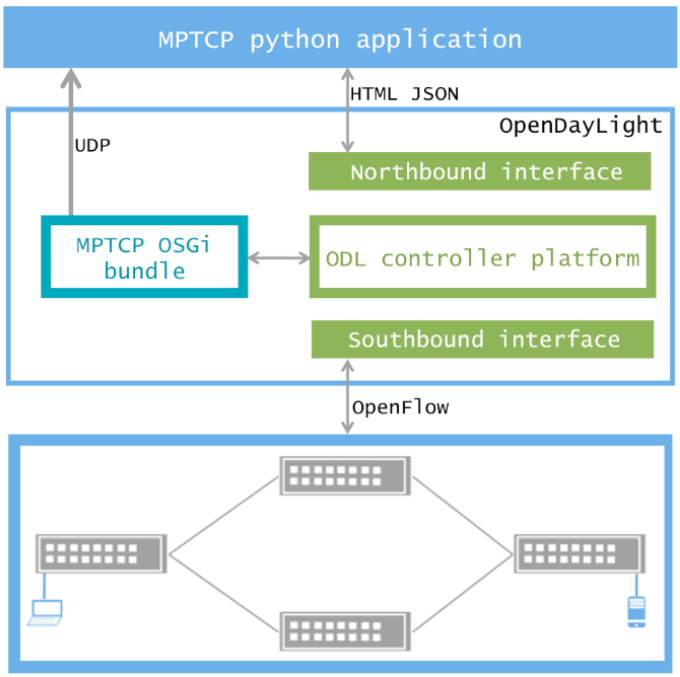

Figure 7.1: General SDN Architecture with MPTCP python application for path management, and OpenDaylight controller with the added MPTCP OSGI module that sends notification to the MPTCP application.

libraries and the flexibility of reactive flow programming. To solve this challenge, we combined both OSGi and RESTful interfaces in a single application-suite that comprise an OSGi bundle for packet-processing logic and a Python path-manager NetApp, running on top of the controller.

The MPTCP OSGi bundle (implemented in JAVA programming language) is the module that runs inside the controller and receives packet-in events. It parses and identifies MPTCP packets to send them to the MPM application for flow programming. Each MPTCP packet is encapsulated in a JSON payload along with the switch-id and connector-id from which the packet came in. The JSON message is sent via a UDP socket to the MPM.

The MPM Python NetApp gets the requests from the SDN controller to create paths for MPTCP traffic (via the UDP socket towards the MPTCP OSGi component running inside the ODL controller). It also uses RESTful APIs of the ODL northbound interfaces to get the topology of the network and saves the topology in a graph library. MPM perform path computation for each request in such a way that individual sub-flows of MPTCP connection take a disjoint path. MPM uses Differential Delay Constrained K Edge-Disjoint Path (DDCKDP)

based algorithms to find least delay paths or least differential delay paths for MPTCP subflows. The algorithm is implemented in MPM using heuristic SPLIT-DDCKDP (Shared Protection of the Largest Individual Traversed link) method as proposed in [269]. Finally, using ODL controller's northbound flow create APIs, MPM issues the instructions to the ODL controller to create flows in the switches using OpenFlow.

- The path engine gets the topology of the network via ODL's northbound interface, it uses Python Networkx library to save the network topology as a graph.
- The MPTCP connection-request packet contains various MPTCP flags including those which instructs the MPM about how many subflows to create for each session. Each subflow for a given connection has different TCP source port number.
- MPM runs a heuristic algorithm that finds the largest set of edge-disjoint paths that meet a prefixed constrain on the differential delay. The computed paths are sorted in order of delay and orderly assigned to the subpaths.
- For each switch in the path, MPM creates flows using ODL northbound flow programming APIs and save the path in an MPTCP connection table.
- When a connection terminates, flows are removed from the switches after idle timeout period.

The great advantage of our MPM implementation is that this NetApp, which can run on a separate server from the one hosting the ODL controller, can potentially act as a single path-computation engine for multiple controllers in a multi-domain network scenario as a module of an orchestrator. That makes our approach suitable also for Transport SDN [38].

7.4 Test-bed implementation

MPTCP Test Manager (TM) is a standalone python application that uses Mininet library to emulate the network and the hosts and to perform MPTCP data transfer between hosts. It takes as input a configuration file that defines all the parameters for a given test case. Multiple test cases can be defined to be executed in a batch, each with a different combination of parameter values for link speed, link delay, scheduler etc.

TM is responsible for the simulation of network switches and hosts, initiation of MPTCP connection from one host to another and the measurements and analysis of the MPTCP connection. It is not responsible for the selection of paths or the setup of flows in switches. MPM handles that part. TM simply acts as data plane and for each new MPTCP connection request or subflow add request, sends an openflow message to SDN controller.

For each test case, Mininet emulates a multipath network and configures the parameters like link bandwidth and link delay for each link according to the values in the configuration file. It then configures MPTCP stack parameters using Linux sysctl commands [270]. MPTCP stack can send traffic from multiple subflows on multiple interfaces of the host machine or on single interface but with different TCP source port number. In our test bed, hosts have single interface and MPTCP multiplexes traffic from different subflows using TCP source port numbers. Once the network is configured properly, the TM starts MPTCP data transfer from simulated hosts using secure copy [271] or iperf [272] and starts taking packet capture file for analysis afterwards. Each subflows of a connection uses different TCP source port number and MPM creates different path for different subflows depending upon the source port. MPM creates flows on switches that include TCP source port and destination port as match parameters and the switch can then route traffic from different subflows over different path. It can also inject background traffic on each link to evaluate the performance of MPTCP in presence of other traffic. After file transfer finishes, TM records the packet capture sequence of each test case and performs advanced analysis to measure a set of performance parameters. Finally, it saves the measures in an easy-to-read CSV file.

7.5 Results

We evaluated MPTCP performance over a simple SDN network with a topology that allowed us exploiting path diversity. In the virtualized test network each link connecting one switch to another has the same fixed capacity, while links have different preconfigured propagation delays. We first tested standard TCP to evaluate a benchmark throughput value, then we repeated the test with MPTCP, using path selection with and without the disjointess condition. In our tests, we have investigated MPTCP in the case of the two-subflows. With link disjoint path selection; there is a significant improvement in the overall throughput. With two disjoint subflows we almost double the throughput compared to single TCP connection.

As mentioned previously, the delay difference of the MP set (differential delay: DD)[1] can reduce the throughput of the overall connection. To better understand this effect, we evaluated the throughput of MPTCP connection using different values of differential delay for two separate test cases: short flows and elephant flows. We recall that each subflow is subject to regular congestion control. Thus, we can define short flows as those connections that live for a short duration and terminates before MPTCP exits slow start phase of congestion control. Elephant flows are long-lived connections that reach the maximum capacity of the link and enters the congestion avoidance state.

[1] DD is defined as the vector containing the differences in delay between the longer and all the other paths belonging to the MP set.

Fig. 7.2 and Fig. 7.3 show the effect of DD between two sub-paths on total throughput and traffic distribution over two flows under two scenarios: short flows and elephant flows, respectively. In this experiment the link capacity has been set to 500 kbit/s. Both cases have been tested using the Lowest RTT first scheduler. Our experiments show that in general we get better throughput when the delay in paths is kept as low as possible, as expected. However, for short-lived flows the dependence is stronger (Fig. 7.2). Slow-start is part of the congestion control strategy used by MPTCP and has an exponential growth phase. The rate of growth is determined by round-trip time, and thus by delay. If paths have different delay, then each subflow exits slow-start phase at different time instants. As shown in the figure, the connection terminates before both subflows can reach the full capacity. The subflow with the highest delay achieve less throughput due to a more brief slow-start progression. For elephant flows (Fig. 7.3), the shorter subflow of the MPTCP connection always reaches the maximum capacity of the link, so the slow start phase does not affect the overall throughput that much.

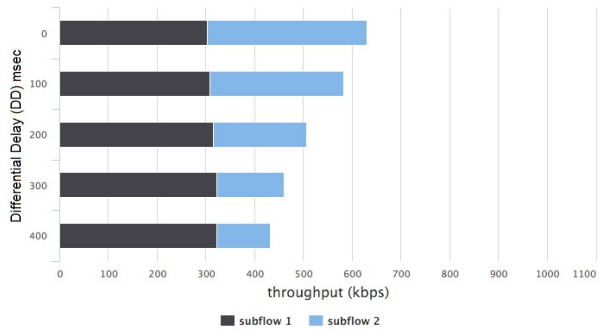

FIGURE 7.2: Differential delay impact on throughput decomposition for short flows connections (MPTCP path engine).

To better understand how delay (d) and DD combine to affect the overall throughput of MPTCP connection, we must consider the effect of path delay for various values of DD and vice versa. We conducted experiments on our automated testbed with two subflows for each MPTCP connection over two disjoint paths each with link bandwidth of 600 kbit/s and tested MPTCP data transfer using a large file to use elephant flows. The maximum throughput without any delays on path was around 1200 kbit/s for MPTCP connection and around 600 kbit/s for TCP connection. We then introduced incremental delays on subflows and tested all possible combination of delay values on different subflows.

Fig. 7.4 and 7.5. 5 show the effect of d and DD on the overall throughput for RR and default scheduler, respectively. Delta curve shows the throughput when the paths experience a variation in DD by setting to zero the delay of

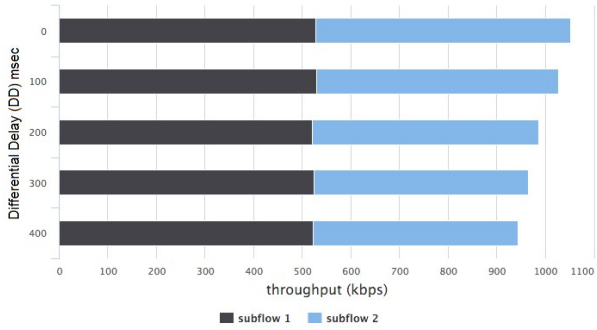

Figure 7.3: Differential delay impact on throughput decomposition for Elephanft flows (MPTCP path engine).

one subflow while varying the delay of the other path according to DD value. Delay curve shows the throughput when both subflows are set to the same value of delay d (DD always equals to 0).

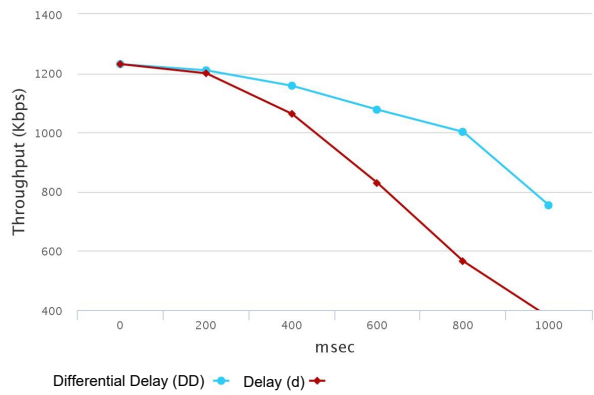

Figure 7.4: Effect of differential delay and delay for RR scheduler.

Fig. 7.6 and Fig. 7.7 show the combined effect of d and DD for RR and default scheduler respectively. Here, subflow 1 is set to a delay value of d and subflow 2 is set to a value, higher or lower than subflow 1, as per DD. For example, the red curve is for d = 400 ms, and the first point which corresponds to a DD of zero means both subflow1 and subflow 2 were set to delay value 400 ms. On this curve, the next point, for DD = 200 ms, implies subflow 2 delay was set to either 200 ms or 600 ms. Throughput values shown on graph is the average throughput for all possible delay values of subflow 2 for a particular DD. For the above point, it is average throughput of connection when delay

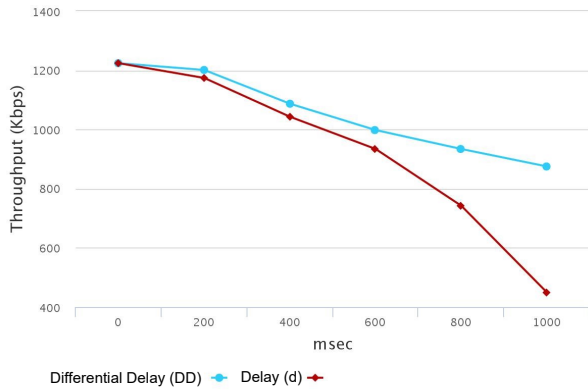

FIGURE 7.5: Effect of differential delay and delay for default scheduler (lowest round trip time first).

on subflow 2 was 200 ms and when the delay on subflow 2 was 600 ms.

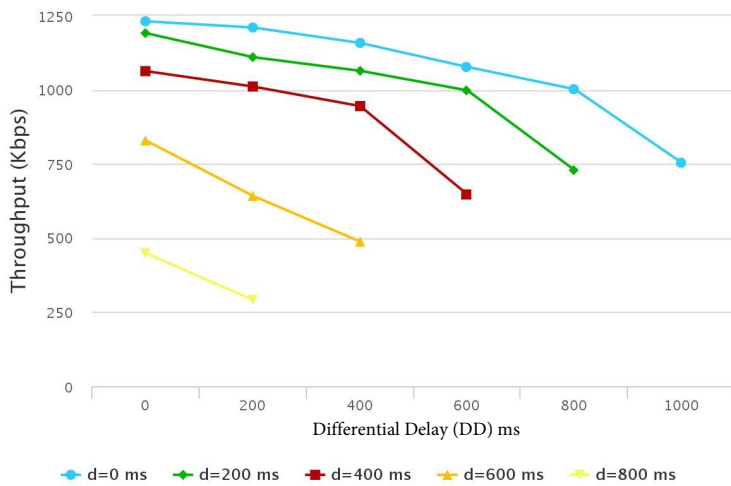

FIGURE 7.6: Combined effect of differential delay and delay for RR scheduler.

RR scheduler gives higher throughput than default scheduler for lower values of DD. This shows that RR can utilize the links much more efficiently than default scheduler when there is differential delay on multiple subflows. However, for higher values of DD, the throughput for RR scheduler decreases considerably compared to default scheduler. As DD increases, buffer overflows at the receiver increases retransmissions.

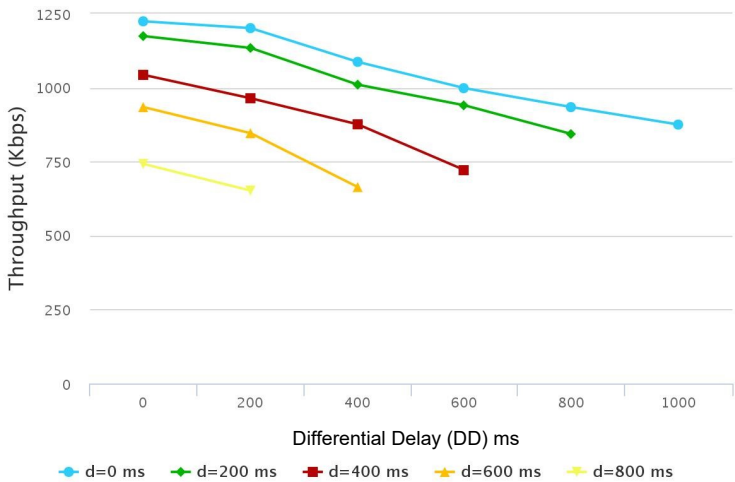

Figure 7.7: Combined effect of differential delay and delay for default scheduler (lowest round trip time first).

Default scheduler may not fully utilize the capacity of all subpaths in the presence of DD, but it performs much better than RR scheduler for higher values of d or DD. This is because the scheduling decision is based on Lowest RTT first principle and for high values of DD; default scheduler can distribute more traffic on better quality paths and low traffic on inferior paths.

7.6 Concluding Remarks

Multipath TCP is one of the most significant changes to TCP. In this Chapter, we presented a flexible SDN-based MPTCP path manager capable of calculating and configuring link disjoint multi-paths for MPTCP connections. We implemented an automated testbed to evaluate MPTCP performance for a wide range of delay values on each subflows. Our automated testbed helps us perform hundreds of test cases consecutively by taking a set of input parameters for the network and gives detailed results of throughput and the distribution of traffic on each subflow. We experimentally demonstrated the effect of delay and differential delay on the performance of MPTCP connections for Lowest RTT first and Round Robin scheduler. Our experiments show that the former performs better for jitter intolerable application but does not utilize the network resources to its maximum capacity. On the other hand, RR scheduler can get higher utilization but at a cost of increased jitter. Our results show exactly how much throughput delay and differential delay affect when both act together.

There are many factors affecting the performance of MPTCP connection. Though we have focused primarily on path diversity in this Chapter, we now

have proper network infrastructure and tools to evaluate other major parameters. In future, we will inspect other factors like congestion control algorithms, buffer size, etc. Moreover, we are working on a real SDN networks to validate the findings of our results.

Concluding Remarks 8

In this thesis we investigated advance optical routing techniques to overcome the static nature and overprovisioning of today's transport networks, to help communication service providers improving their revenue margin that paradoxically has been dropping with the Internet traffic explosion. Our work led us to explore the SDN concept, which is quickly reshaping the scientific and industrial networking world, unleashing programmability and dynamism that previously seemed to be impossible, and today become the basis for advance routing techniques and a myriad of new services. We elaborated on two advanced routing techniques that benefit from such programmability: dynamic optical routing and multipath routing. Finally, we demonstrated the implementation of SDN in a brown-field transport scenario proposed by TIM (Telecom Italia), and a unique SDN-based automated test-bed for multipath TCP in a green-field scenario.

The transport network provider must control multi-domain and multi-vendor network in some cases composed by diverse optical technologies, that challenged the model of SDN as it was conceived for purely-packet and homogeneous networks. In consequence, as we presented in Chapter 2 SDN had to be reviewed before extending it to transport networks (T-SDN). Previous works that surveyed T-SDN have focused mainly in research efforts. In this thesis we successfully analysed the whole ecosystem of transport networks including academia, industry, standardization and open source projects, investigating the topic under the point of view of large transport-network operators such as TIM (the Italian incumbent operator).

The process of effectively matching SDN and optical networks using a uniform abstraction was the first step towards T-SDN (SDON). However, this is still an ongoing challenge because optical-equipment vendors have added value to their solutions by introducing innovative features and capacities that

differentiate their products from other vendors. A solution to this heterogeneity is the hierarchical control plane (HT-SDN) paradigm. In HT-SDN domain-specific controllers provide abstracted views towards higher order controllers or network orchestrator. HT-SDN allows the co-existence of SDN with other legacy but widespread control-plane implementations, such as GMPLS. So, it is a key to accelerate T-SDN deployment. Transport-network orchestration must be ideally vendor agnostic, avoiding vendor lock-in supported by standard NBIs. Thus, HT-SDN moves the issue of standardization from SBI to NBI.

An important evolutionary step (already happening in other segments) is the integration of SDN and NFV to foster the deployment of control plane and virtual data-plane network functionalities, adding all features (e.g. security, caching, optimization) to provide services to final customers (business, residential, mobile, etc.). The last evolutionary step is given by the use of disaggregated white-box ROADM, that marks the inflexion point between closed and vendor-specific optical devices and the embrace of COTS optical networking equipment.

After the deep dive in T-SDN, we assumed an scenario in which T-SDN is already deployed to focus on advanced optical routing techniques that take advantage of programmability, visibility and centralized control of transport networks.

In Chapter 3 we proposed dynamic optical routing techniques to avoid resource overprovisioning and reduce the energy consumption of a particular transport network use case: the mobile metro-core networks. While related works are mainly focused on time-dependent overall traffic demand, in this work we showed that predictable spatio-temporal traffic demand fluctuations (tidal traffic) are a valuable information for optimization of dynamic optical routing techniques. After demonstrating the predictability of tidal traffic, we proposed two off-line optimization models for predictable traffic conditions that reported daily energy savings of more than 20%. When comparing our method based on load adaptive network operation and optical bypass with the static approach our results show savings up to 47.5%. For unpredictable traffic load conditions, we proposed two novel matheuristic methods that exploit metropolitan tidal traffic predictability to generate optimal weights using off-line models, and then reducing the problem to a link-disjoint path-pairs problem. The matheuristics reached an optimality gap below 1.5%, with an improvement of 8.5% when compared to a matheuristic that do not exploits the tidal traffic predictability. We have also proposed a scheduling heuristic that provides a very good trade-off with a large reduction of routing changes (reconfiguration time points) and a small penalty on energy savings (resource allocation efficiency).

In Chapter 4 we start exploring the main challenge of another advanced optical routing technique: the differential delay (DD) experienced in multipath routing. Assuming transport network programmability, and centralized path computation we propose two optimization models that included the path-disjointness constraint into the well known DD-constrained MP routing problem (3D; Disjoint Diff. Delay). We provided an iterative optimization approach

that makes 3D problem tractable. We introduced a novel technique that allows controlled in-path routing cycles to equalize the paths' delays, and as a consequence broadens the solution space of the problem. However, for both formulations we evidenced a trade-off between DD minimization and average end-to-end delay of the MP connection. In fact, this is a general problem of DD-constrained MP routing, because in order to equalize the paths delay, the solutions may be much longer than the shortest path.

The trade-off identified in 4 led us to investigate DD-compensated (DDC) MP routing for optical transport networks in Chapter 5. To the best of our knowledge, we proposed for the first-time an FDL-based transparent-DDC technique. Transparent-DDC allows to reduce the impact of DD in MP routing over optical networks, without the limitations of electronic reconstruction buffers. Commercially available rack mount FDL-modules [241], make our proposal very interesting for real implementation of transparent-DDC capabilities in optical nodes. We proposed hybrid-DDC optimization models that mixed electronic-DDC and transparent-DDC Numerical results showed that: transparent-DDC avoids energy consuming O/E-E/O conversions and reduces the size of expensive reconstruction buffers. Moreover we demonstrated that the discrete nature of FDLs does not impact the performance of hybrid-DDC models. It is interesting to notice that to further avoid O/E-E/O conversions, there is an optimum hybrid-DDC model that should be selected depending on the network dimension.

Following the general architectural knowledge got in Chapter 2 and the experience on routing techniques from chapters 5 and 4, we have demonstrated the implementation of two SDN use cases in: a brown-field (MPLS network) and a green-field (OpenFlow network) virtualized scenarios.

In Chapter 6 we demonstrated the first standard-base hierarchical SDN control-plane architectures for a Segment Routing (SR) multi-domain network, using OpenDaylight as domain controllers and a network orchestration on top of them. This work demonstrated for the first-time standard methods to exchange SR segment ID (SID) information in a multi-domain scenario. We showed the use of multi path technique supported for load balancing and application-aware path selection to guarantee QoS. SR proved to be useful for enabling T-SDN in brown-field scenarios by adding the proper south bound interfaces (BGP-LS and PCEP) to the edge of (G)MPLS networks. The HT-SDN architecture enables network providers to add new value to their business as they can dynamically address new requirements from customers.

Finally, in Chapter 7, we presented a flexible SDN-based multipath TCP (MPTCP) path manager and the related automated testbed over a green-field OpenFlow network. MPTCP is one of the most significant changes to TCP, and SDN proved to support innovation. The path manager is an application on top of a SDN controller, that is capable of calculating and configuring link disjoint multi-paths for MPTCP connections, to enhance its performance. We implemented an automated testbed to evaluate MPTCP performance that can be affected by many factors. Using the automated testbed tool, we

experimentally demonstrated the performance degradation of MPTCP due to delay, DD and the MPTCP schedulers.

Although the presented research is at its initial stage, we believe that the shown optimization problems and algorithms are enough general and powerful to serve as basis for possible extensions, in order to study some interesting open issues that emerged and were discussed in this work.

The dynamic optical routing framework will be further extended with advance prediction techniques, and can be implemented as applications that run on top of and SDN orchestrator. Moreover an SDN-NFV architecture based on the ONOS-CORD [32] will be use to extend the work to virtual evolve packet core (vEPC) and video content caching.

Regarding the transparent-DDC, an on-line methodology will be an interesting work, that can lead to the use of transparent-DDC in dynamic optical routing techniques.

The SDN implementations serve as basis for future developments, for instance the HT-SDN Orchestrator can be extended to support other open source controllers such as ONOS, and provide scheduled traffic engineering use cases like bandwidth calendaring. In the MPTCP automated testbed we have focused primarily on path diversity and MPTCP scheduler, we now have proper network infrastructure and tools to evaluate other major parameters such as congestion control algorithms and buffer size.

During this project we were not able to test networks involving the optical layer due to the lack of an optical network layer either real (to expensive) or virtualized/emulated (no open source tools). Thus, a very useful work that should be done is the development of an emulated optical network layer.

T-SDN is a reality: it cannot be clearer that there is a huge demand for T-SDN. However, T-SDN is just in an initial stage, and there are many open issues to be solved. There is a long path before stable standards will rule the implementation of T-SDN. Therefore, it is really fascinating and exciting for researchers to be part of this evolution, but - more important - economically vital for transport-network operators. In the mean time, SDN allows to create rapid prototyping to test innovative ideas in virtualized environments or even better in an isolated network slice. Which represents a unique tool for new developments pushed by academic researchers.

The pursue of this PhD was an enriching journey that boosted my professional career and personal growth. I had the opportunity to share personal and professional experiences with students, researchers, professors, vendors and service providers; enlarging my connections in the networking area all around the world. I tutored master students from Italy, Serbia, China, Colombia and Venezuela. I went to the Beijing University of Post and Telecommunications (China), where we started a collaboration project with a Chinese research group, that is still ongoing. I have taught in Italian two different courses for the undergrad students at Politecnico di Milano, and I am now able to discuss technical and non-technical matters in Italian.

Bibliography

[1] S. Azodolmolky *et al.*, "Integrated OpenFlow-GMPLS control plane: an overlay model for software defined packet over optical networks," *Opt. Express*, vol. 19, no. 26, pp. B421–B428, Dec 2011.

[2] L. Liu *et al.*, "OpenFlow-based Wavelength Path Control in Transparent Optical Networks: a Proof-of-Concept Demonstration," in *European Conference and Exposition on Optical Communications (ECOC)*. Optical Society of America, 2011, p. Tu.5.K.2.

[3] M. Siqueira *et al.*, "An optical SDN Controller for Transport Network virtualization and autonomic operation," in *IEEE Globecom Workshops*, Dec 2013, pp. 1198–1203.

[4] M. Bjorklund, "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)," RFC 6020, Oct. 2015. [Online]. Available: https://rfc-editor.org/rfc/rfc6020.txt

[5] Y. Zhao *et al.*, "Unified control system for heterogeneous networks with Software Defined Networking (SDN)," in *International ICST Conference on Communications and Networking in China (CHINACOM)*, Aug 2013, pp. 781–784.

[6] A. Mayoral *et al.*, "Experimental validation of automatic lightpath establishment integrating OpenDayLight SDN controller and Active Stateful PCE within the ADRENALINE testbed," in *International Conference on Transparent Optical Networks (ICTON)*, July 2014, pp. 1–4.

[7] R. Casellas *et al.*, "SDN orchestration of OpenFlow and GMPLS flexi-grid networks with a stateful hierarchical PCE [invited]," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 7, no. 1, pp. A106–A117, Jan 2015.

[8] R. Munoz *et al.*, "Experimental assessment of ABNO-based network orchestration of end-to-end multi-layer (OPS/OCS) provisioning across SDN/OpenFlow and GMPLS/PCE control domains," in *European Conference on Optical Communication (ECOC)*, Sept 2014, pp. 1–3.

[9] ONF-OTWG, "OpenFlow-enabled Transport SDN," ONF Solution Brief, May 2014. [Online]. Available: https://www.opennetworking.org

[10] Optical Internetworking Forum, "Framework for Transport SDN: Components and APIs," OIF-FD-TRANSPORT-SDN-01.0, May 2015. [Online]. Available: http://www.oiforum.com/

[11] D. King and A. Farrel, "A PCE-Based Architecture for Application-Based Network Operations," RFC 7491, Mar. 2015. [Online]. Available: https://rfc-editor.org/rfc/rfc7491.txt

[12] (2016) Cisco Systems. [Online]. Available: http://www.cisco.com/

[13] G. Cook and J. Van Horn, "How dirty is your data? a look at the energy choices that power cloud computing," *Greenpeace International*, 2011.

[14] D. Kreutz *et al.*, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, Jan 2015.

[15] (2015) Open Networking Foundation. [Online]. Available: http://www.opennetworking.org

[16] Open Networking Foundation, "Software-defined networking: The new norm for networks," *ONF White Paper*, Apr 2012.

[17] Optical Internetworking Forum and Open Networking Foundation, "Global Transport SDN Prototype Demonstration," OIF/ONF White Paper, Oct 2014. [Online]. Available: http://www.oiforum.com/

[18] A. Lara, A. Kolasani, and B. Ramamurthy, "Network Innovation using OpenFlow: A Survey," *IEEE Communications Surveys Tutorials*, vol. 16, no. 1, pp. 493–512, First 2014.

[19] S. Jain *et al.*, "B4: Experience with a Globally-deployed Software Defined Wan," *SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, pp. 3–14, Aug 2013.

[20] C. Lange and A. Gladisch, "Energy Efficiency Limits of Load Adaptive Networks," in *Optical Fiber Communication Conference (OFC)*, 2010, p. OWY2.

[21] Ericson, "Ericson Mobiliy Report," 2016. [Online]. Available: http://www.ericsson.com

[22] Mobile and wireless communications Enablers for the Twenty-twenty Information Society (METIS), "Channel Models," ICT-317669-METIS/D1.4, 2015.

[23] C. Song *et al.*, "Limits of Predictability in Human Mobility," *Science*, vol. 327, no. 5968, pp. 1018–1021, 2010.

[24] Z. Niu, "Tango: traffic-aware network planning and green operation," *IEEE Wireless Commun.*, vol. 18, no. 5, pp. 25–29, October 2011.

[25] S. K. Singh, T. Das, and A. Jukan, "A survey on internet multipath routing and provisioning," *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 2157–2175, Fourthquarter 2015.

[26] T. D. Wallace and A. Shami, "On-demand scheduling for concurrent multipath transfer under delay-based disparity," in *2012 8th International Wireless Communications and Mobile Computing Conference (IWCMC)*, Aug 2012, pp. 833–837.

[27] A. Anurag Srivastava *et al.*, "Differential delay aware routing for Ethernet over SONET/SDH," in *IEEE INFOCOM*, vol. 2, Mar. 2005, pp. 1117–1127.

[28] S. Ahuja, T. Korkmaz, and M. Krunz, "Minimizing the differential delay for virtually concatenated Ethernet over SONET systems," in *IEEE ICCCN*, Oct. 2004, pp. 205–210.

[29] R. Alvizu *et al.*, "Differential delay constrained multipath routing for sdn and optical networks," *Elsevier Electronic Notes in Discrete Mathematics*, vol. 52, pp. 277–284, Jun 2016.

[30] M. Alicherry, C. Phadke, and V. Poosala, "Delay distributed VCAT for efficient data-optical transport," in *OFC*, vol. 4, Mar. 2005, p. 3 pp.

[31] (2015) OpenDaylight Project. [Online]. Available: http://www.opendaylight.org

[32] (2015) Open Network Operating System Foundation. [Online]. Available: http://onosproject.org/

[33] F. Hu, Q. Hao, and K. Bao, "A Survey on Software-Defined Network and OpenFlow: From Concept to Implementation," *IEEE Communications Surveys Tutorials*, vol. 16, no. 4, pp. 2181–2206, Fourthquarter 2014.

[34] H. Farhady, H. Lee, and A. Nakao, "Software-Defined Networking," *Comput. Netw.*, vol. 81, no. C, pp. 79–95, Apr 2015.

[35] A. Blenk *et al.*, "Survey on Network Virtualization Hypervisors for Software Defined Networking," *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 655–685, Firstquarter 2016.

[36] S. M. *et al.*, "OpenFlow and Multi-layer Extensions: Overview and Next Steps," in *Software Defined Networking (EWSDN), 2012 European Workshop on*, Oct 2012, pp. 13–17.

[37] J.-P. Elbers and A. Autenrieth, "From static to software-defined optical networks," in *Optical Network Design and Modeling (ONDM), 2012 16th International Conference on*, April 2012, pp. 1–4.

[38] R. Alvizu and G. Maier, "Can open flow make transport networks smarter and dynamic? An overview on transport SDN," in *Smart Communications in Network Technologies (SaCoNeT), 2014 International Conference on*, June 2014, pp. 1–6.

[39] J. R. de Almeida Amazonas, G. Santos-Boada, and J. Sole-Pareta, "A critical review of OpenFlow/SDN-based networks," in *Transparent Optical Networks (ICTON), 2014 16th International Conference on*, July 2014, pp. 1–5.

[40] G. Zhang *et al.*, "A survey on ofdm-based elastic core optical networking," *IEEE Communications Surveys Tutorials*, vol. 15, no. 1, pp. 65–87, First 2013.

[41] A. Thyagaturu *et al.*, "Software Defined Optical Networks (SDONs): A Comprehensive Survey," *IEEE Communications Surveys Tutorials*, vol. PP, no. 99, pp. 1–1, 2016.

[42] R. Casellas *et al.*, "Overarching Control of Flexi Grid Optical Networks: Interworking of GMPLS and OpenFlow Domains," *Journal of Lightwave Technology*, vol. 33, no. 5, pp. 1054–1062, March 2015.

[43] A. Martinez and M. Yannuzzi and V. López and D. López and W. Ramírez and R. Serral-Gracià and X. Masip-Bruin and M. Maciejewski and J. Altmann, "Network Management Challenges and Trends in Multi-Layer and Multi-Vendor Settings for Carrier-Grade Networks," *IEEE Communications Surveys Tutorials*, vol. 16, no. 4, pp. 2207–2230, Fourthquarter 2014.

[44] N. Cvijetic, "SDN for Optical Access Networks," in *Advanced Photonics for Communications*. Optical Society of America, 2014, p. PM3C.4. [Online]. Available: http://www.osapublishing.org/abstract.cfm?URI=PS-2014-PM3C.4

[45] N. Cvijetic *et al.*, "SDN and OpenFlow for Dynamic Flex-Grid Optical Access and Aggregation Networks," *Journal of Lightwave Technology*, vol. 32, no. 4, pp. 864–870, Feb 2014.

[46] L. Liu, "Sdn orchestration for dynamic end-to-end control of data center multi-domain optical networking," *China Communications*, vol. 12, no. 8, pp. 10–21, August 2015.

[47] R. A. G. Maier *et al.*, "The big challenge: Software Defined Networking for Transport Data Networks," *submitted for publication to IEEE Communication Surveys and Tutorials*.

[48] Open Networking Foundation, "OpenFlow Switch Specification. Version 1.4.0 (Wire Protocol 0x05)," ONF Technical Specification-012, Oct 2013. [Online]. Available: https://www.opennetworking.org/

[49] ONF-OTWG. (2015, Mar) Optical Transport Protocol Extensions. V1.0. [Online]. Available: https://www.opennetworking.org

[50] ONF-OTWG. (2014, Aug) Requirements Analysis for Transport OpenFlow/SDN. V1.0. [Online]. Available: https://www.opennetworking.org

[51] ONF-OTWG. (2015, Oct) Wireless Transport SDN Proof of Concept White Paper. V1.0. [Online]. Available: https://www.opennetworking.org

[52] E. Rosen and R. Callon, "Multiprotocol Label Switching Architecture," RFC 3031, Mar. 2013. [Online]. Available: https://rfc-editor.org/rfc/rfc3031.txt

[53] D. Frost, L. Levrau, and M. Bocci, "MPLS Transport Profile User-to-Network and Network-to-Network Interfaces," RFC 6215, Oct. 2015. [Online]. Available: https://rfc-editor.org/rfc/rfc6215.txt

[54] *Interfaces for Optical Transport Network*, G.709, ITU-T Std., Feb 2012.

[55] *Spectral grids for WDM applications: DWDM frequency grid*, Recommentation G.694.1, ITU-T Std., Feb 2012.

[56] I. Tomkos *et al.*, "A tutorial on the flexible optical networking paradigm: State of the art, trends, and research challenges," *Proceedings of the IEEE*, vol. 102, no. 9, pp. 1317–1337, Sept 2014.

[57] O. Gerstel *et al.*, "Elastic optical networking: a new dawn for the optical layer?" *Communications Magazine, IEEE*, vol. 50, no. 2, pp. s12–s20, Feb 2012.

[58] E. Mannie, "Generalized Multi-Protocol Label Switching (GMPLS) Architecture," RFC 3945, Mar. 2013. [Online]. Available: https://rfc-editor.org/rfc/rfc3945.txt

[59] A. Farrel and G. R. Ash, "A Path Computation Element (PCE)-Based Architecture," RFC 4655, Oct. 2015. [Online]. Available: https://rfc-editor.org/rfc/rfc4655.txt

[60] R. Doverspike and J. Yates, "Optical network management and control," *Proceedings of the IEEE*, vol. 100, no. 5, pp. 1092–1104, May 2012.

[61] N. McKeown *et al.*, "OpenFlow: Enabling Innovation in Campus Networks," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, Mar. 2008.

[62] Open Networking Foundation, "SDN/OpenFlow Products," Open Networking Foundation report, 2015. [Online]. Available: https://www.opennetworking.org

[63] S. Das, G. Parulkar, and N. McKeown, "Simple unified control for packet and circuit networks," in *IEEE/LEOS Summer Topical Meeting (LEOSST)*, July 2009, pp. 147–148.

[64] B. Mukherjee, *Optical WDM Networks*. Springer-Verlag NY, Inc., 2006.

[65] B. Collings, "New devices enabling software-defined optical networks," *IEEE Communications Magazine*, vol. 51, no. 3, pp. 66–71, March 2013.

[66] R. Younce, J. Larikova, and Y. Wang, "Engineering 400g for colorless-directionless-contentionless architecture in metro/regional networks [invited]," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 5, no. 10, pp. A267–A273, Oct 2013.

[67] N. Amaya, G. Zervas, and D. Simeonidou, "Introducing node architecture flexibility for elastic optical networks," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 5, no. 6, pp. 593–608, June 2013.

[68] M. Garrich *et al.*, "Experimental demonstration of function programmable add/drop architecture for roadms [invited]," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 7, no. 2, pp. A335–A343, February 2015.

[69] P. Bhaumik *et al.*, "Software-defined optical networks (sdons): a survey," *Photonic Network Communications*, vol. 28, no. 1, pp. 4–18, 2014.

[70] Optical Internetworking Forum, "OIF Carrier WG Requirements on Transport Networks in SDN Architectures," OIF Technical Document, Sep 2013. [Online]. Available: http://www.oiforum.com/

[71] Z. Zhu *et al.*, "Demonstration of cooperative resource allocation in an openflow-controlled multidomain and multinational sd-eon testbed," *J. Lightwave Technol.*, vol. 33, no. 8, pp. 1508–1514, Apr 2015.

[72] S. Das, G. Parulkar, and N. McKeown, "Unifying Packet and Circuit Switched Networks," in *IEEE GLOBECOM Workshops*, Nov 2009, pp. 1–6.

[73] S. Das, "Extensions to the OpenFlow protocol in support of circuit switching," *Addendum to OpenFlow protocol specification (v1. 0) - Circuit Switch Addendum v0*, vol. 3, 2010.

[74] V. R. Gudla *et al.*, "Experimental Demonstration of OpenFlow Control of Packet and Circuit Switches," in *Optical Fiber Communication Conference*. Optical Society of America, 2010, p. OTuG2.

[75] S. Das *et al.*, "Packet and Circuit Network Convergence with OpenFlow," in *Optical Fiber Communication Conference*. Optical Society of America, 2010, p. OTuG1.

[76] S. Das *et al.*, "Application-Aware Aggregation and Traffic Engineering in a Converged Packet-Circuit Network," in *Optical Fiber Communication Conference*. Optical Society of America, 2011, p. NThD3.

[77] S. Das. (2011) Aggregation on a Converged Packet-Circuit Network. [Online]. Available: http://openflowswitch.org/wk/index.php/Aggregation_on_a_Converged_Packet-Circuit_Network

[78] D. Simeonidou, R. Nejabati, and S. Azodolmolky, "Enabling the future optical Internet with OpenFlow: A paradigm shift in providing intelligent optical network services," in *International Conference on Transparent Optical Networks (ICTON)*, June 2011, pp. 1–4.

[79] L. Liu *et al.*, "Experimental validation and performance evaluation of OpenFlow-based wavelength path control in transparent optical networks," *Opt. Express*, vol. 19, no. 27, pp. 26 578–26 593, Dec 2011.

[80] L. Liu, T. Tsuritani, and I. Morita, "Experimental demonstration of OpenFlow/GMPLS interworking control plane for IP/DWDM multi-layer optical networks," in *International Conference on Transparent Optical Networks (ICTON)*, July 2012, pp. 1–4.

[81] M. Channegowda *et al.*, "Experimental Evaluation of Extended OpenFlow Deployment for High-Performance Optical Networks," in *European Conference and Exhibition on Optical Communication*. Optical Society of America, 2012, p. Tu.1.D.2.

[82] L. Liu *et al.*, "First Field Trial of an OpenFlow-based Unified Control Plane for Multi-layer Multi-granularity Optical Networks," in *Optical Fiber Communication Conference (OFC)*, Mar 2012, p. PDP5D.2.

[83] L. Liu *et al.*, "OpenSlice: An OpenFlow-based control plane for spectrum sliced elastic optical path networks," in *European Conference and Exhibition on Optical Communications (ECOC)*, Sept 2012, pp. 1–3.

[84] L. Liu *et al.*, "Interworking between OpenFlow and PCE for dynamic wavelength path control in multi-domain WSON," in *Optical Fiber Communication Conference (OFC)*, Mar 2012, pp. 1–3.

[85] S. Azodolmolky *et al.*, "Optical FlowVisor: An OpenFlow-based Optical Network Virtualization Approach," in *National Fiber Optic Engineers Conference*. Optical Society of America, Mar 2012, p. JTh2A.41.

[86] S. Das, G. Parulkar, and N. McKeown, "Rethinking IP Core Networks," *J. Opt. Commun. Netw.*, vol. 5, no. 12, pp. 1431–1442, Dec 2013.

[87] M. Channegowda *et al.*, "Experimental demonstration of an OpenFlow based software-defined optical network employing packet, fixed and flexible DWDM grid technologies on an international multi-domain testbed," *Opt. Express*, vol. 21, no. 5, pp. 5487–5498, Mar 2013.

[88] L. Liu *et al.*, "Demonstration of a Dynamic Transparent Optical Network Employing Flexible Transmitters/Receivers Controlled by an OpenFlow-Stateless PCE Integrated Control Plane [Invited]," *J. Opt. Commun. Netw.*, vol. 5, no. 10, pp. A66–A75, Oct 2013.

[89] H. Y. Choi *et al.*, "Demonstration of BER-Adaptive WSON Employing Flexible Transmitter/Receiver With an Extended OpenFlow-Based Control Plane," *IEEE Photonics Technology Letters*, vol. 25, no. 2, pp. 119–121, Jan 2013.

[90] R. Casellas *et al.*, "An integrated stateful PCE/OpenFlow controller for the control and management of flexi-grid optical networks," in *Optical Fiber Communication Conference (OFC)*, Mar 2013, pp. 1–3.

[91] R. Casellas *et al.*, "Control and management of flexi-grid optical networks with an integrated stateful path computation element and OpenFlow controller [invited]," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 5, no. 10, pp. A57–A65, Oct 2013.

[92] L. Liu *et al.*, "OpenSlice: an OpenFlow-based control plane for spectrum sliced elastic optical path networks," *Opt. Express*, vol. 21, no. 4, pp. 4194–4204, Feb 2013.

[93] J. Oliveira *et al.*, "Experimental testbed of reconfigurable flexgrid optical network with virtualized GMPLS control plane and autonomic controls towards SDN," in *SBMO/IEEE MTT-S International Microwave Optoelectronics Conference (IMOC)*, Aug 2013, pp. 1–5.

[94] M. Bahnasy, K. Idoudi, and H. Elbiaze, "Software-defined DWDM optical networks: OpenFlow and GMPLS experimental study," in *Global Communications Conference (GLOBECOM)*, Dec 2014, pp. 2173–2179.

[95] M. Bahnasy, K. Idoudi and H. Elbiaze, "OpenFlow and GMPLS Unified Control Planes: Testbed Implementation and Comparative Study," *J. Opt. Commun. Netw.*, vol. 7, no. 4, pp. 301–313, Apr 2015.

[96] E. Magalhães *et al.*, "Global roadm-based spectrum equalizer in sdn architecture for qot optimization at dwdm networks," in *Optical Fiber Communications Conference and Exhibition (OFC), 2014*, March 2014, pp. 1–3.

[97] G. C. Santos et al, "Employment of ia-rwa in virtual optical networks using a pce implemented as a sdn application," in *Computer Networks and Distributed Systems (SBRC), 2014 Brazilian Symposium on*, May 2014, pp. 207–213.

[98] J. Oliveira, et al, "Toward terabit autonomic optical networks based on a software defined adaptive/cognitive approach [invited]," *J. Opt. Commun. Netw.*, vol. 7, no. 3, pp. A421–A431, Mar 2015.

[99] M. Siqueira *et al.*, "Providing Optical Network as a Service with Policy-based Transport SDN," *Journal of Network and Systems Management*, vol. 23, no. 2, pp. 360–373, 2015.

[100] A. Mayoral *et al.*, "Integrated IT and network orchestration using OpenStack, OpenDaylight and active stateful PCE for intra and inter data center connectivity," in *European Conference on Optical Communication (ECOC)*, Sept 2014, pp. 1–3.

[101] A. Aguado *et al.*, "ABNO: a feasible SDN approach for multi-vendor IP and optical networks," in *Optical Fiber Communication Conference*, 2014, p. Th3I.5.

[102] Y. Yoshida *et al.*, "First international SDN-based Network Orchestration of Variable-capacity OPS over Programmable Flexi-grid EON," in *Optical Fiber Communication Conference: Postdeadline Papers*. Optical Society of America, 2014, p. Th5A.2.

[103] A. Aguado *et al.*, "ABNO: A Feasible SDN Approach for Multivendor IP and Optical Networks [Invited]," *J. Opt. Commun. Netw.*, vol. 7, no. 2, pp. A356–A362, Feb 2015.

[104] Y. Yoshida *et al.*, "SDN-Based Network Orchestration of Variable-Capacity Optical Packet Switching Network Over Programmable Flexi-Grid Elastic Optical Path Network," *J. Lightwave Technol.*, vol. 33, no. 3, pp. 609–617, Feb 2015.

[105] R. M. noz *et al.*, "Transport Network Orchestration for End-to-End Multilayer Provisioning Across Heterogeneous SDN/OpenFlow and GMPLS/PCE Control Domains," *J. Lightwave Technol.*, vol. 33, no. 8, pp. 1540–1548, Apr 2015.

[106] R. Vilalta, et al., "The need for a Control Orchestration Protocol in research projects on optical networking," in *European Conference on Networks and Communications (EuCNC)*, Jun 2015, pp. 340–344.

[107] A. M. L. de Lerma et al., "First experimental demonstration of distributed cloud and heterogeneous network orchestration with a common transport api for e2e services with qos," in *Optical Fiber Communication Conference*. Optical Society of America, 2016, p. Th1A.2.

[108] A. N. Patel *et al.*, "Distance-adaptive virtual network embedding in software-defined optical networks," in *OptoElectronics and Communications Conference*, June 2013, pp. 1–2.

[109] Z. Ye *et al.*, "Virtual infrastructure embedding over software-defined flex-grid optical networks," in *IEEE Globecom Workshops*, Dec 2013, pp. 1204–1209.

[110] A. Autenrieth *et al.*, "Evaluation of virtualization models for optical connectivity service providers," in *International Conference on Optical Network Design and Modeling*, May 2014, pp. 264–268.

[111] R. Vilalta, et al, "Dynamic Multi-domain Virtual Optical Networks Deployment with Heterogeneous Control Domains," in *Optical Fiber Communication Conference*. Optical Society of America, 2014, p. M3H.4.

[112] J. Zhang, Y. Zhao, and H. Yang, "Software Defined Networking: From dynamic lightpath provisioning to optical virtualization," in *OptoElectronics and Communication Conference*, July 2014, pp. 691–693.

[113] R. Vilalta, et al, "Dynamic Multi-Domain Virtual Optical Network Deployment With Heterogeneous Control Domains [Invited]," *J. Opt. Commun. Netw.*, vol. 7, no. 1, pp. A135–A141, Jan 2015.

[114] R. Vilalta *et al.*, "Network virtualization controller for abstraction and control of openflow-enabled multi-tenant multi-technology transport networks," in *Optical Fiber Communications Conference*, March 2015, pp. 1–3.

[115] Z. Ye *et al.*, "Survivable virtual infrastructure mapping with dedicated protection in transport software-defined networks [invited]," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 7, no. 2, pp. A183–A189, February 2015.

[116] R. Muñoz *et al.*, "SDN/NFV orchestration for dynamic deployment of virtual SDN controllers as VNF for multi-tenant optical networks," in *Optical Fiber Communications Conference (OFC)*, March 2015, pp. 1–3.

[117] A. Aguado *et al.*, "Dynamic virtual network reconfiguration over sdn orchestrated multitechnology optical transport domains," *J. Lightwave Technol.*, vol. 34, no. 8, pp. 1933–1938, April 2016.

[118] K. Bogineni et al, "SDN-NFV Reference Architecture," Version 1.0, feb 2016. [Online]. Available: http://innovation.verizon.com/content/dam/vic/PDF/Verizon_SDN-NFV_Reference_Architecture.pdf

[119] R. Vilalta *et al.*, "Multitenant transport networks with sdn/nfv," *J. Lightwave Technol.*, vol. 34, no. 6, pp. 1509–1515, March 2016.

[120] R. Martínez *et al.*, "Integrated SDN/NFV Orchestration for the Dynamic Deployment of Mobile Virtual Backhaul Networks over a Multi-layer (Packet/Optical) Aggregation Infrastructure," in *Optical Fiber Communication Conference*. Optical Society of America, 2016, p. Th1A.1.

[121] A. Giorgetti *et al.*, "OpenFlow and PCE architectures in Wavelength Switched Optical Networks," in *International Conference on Optical Network Design and Modeling (ONDM)*, April 2012, pp. 1–6.

[122] L. Liu *et al.*, "Field Trial of an OpenFlow-Based Unified Control Plane for Multilayer Multigranularity Optical Switching Networks," *Journal of Lightwave Technology*, vol. 31, no. 4, pp. 506–514, Feb 2013.

[123] (2008) NOX. [Online]. Available: http://www.noxrepo.org/

[124] S. Sivabalan *et al.*, "PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model," Internet Engineering Task Force, Internet-Draft draft-ietf-pce-pce-initiated-lsp-05, Apr. 2016, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-ietf-pce-pce-initiated-lsp-05

[125] (2015) Project Floodlight. [Online]. Available: http://www.projectfloodlight.org

[126] R. Sherwood *et al.*, "Flowvisor: A network virtualization layer," OpenFlow Technical Report TR-2009-1, Oct 2009. [Online]. Available: http://archive.openflow.org/

[127] R. Vilalta *et al.*, "Transport PCE network function virtualization," in *2014 The European Conference on Optical Communication (ECOC)*, Sept 2014, pp. 1–3.

[128] (2016) OPEN-Orchestrator Project. [Online]. Available: http://www.open-o.org

[129] D. Zhang *et al.*, "Highly survivable software defined synergistic ip+optical transport networks," in *Optical Fiber Communication Conference*. Optical Society of America, 2014, p. Th3B.6.

[130] H. Yang *et al.*, "Performance evaluation of multi-stratum resources integrated resilience for software defined inter-data center interconnect," *Opt. Express*, vol. 23, no. 10, pp. 13 384–13 398, May 2015.

[131] L. Liu *et al.*, "Experimental demonstration of OpenFlow-based dynamic restoration in elastic optical networks on GENI testbed," in *European Conference on Optical Communication (ECOC)*, Sept 2014, pp. 1–3.

[132] L. Liu *et al.*, "Dynamic OpenFlow-Based Lightpath Restoration in Elastic Optical Networks on the GENI Testbed," *J. Lightwave Technol.*, vol. 33, no. 8, pp. 1531–1539, Apr 2015.

[133] (2015) GENI (Global Environment for Network Innovations). [Online]. Available: https://www.geni.net

[134] A. Giorgetti *et al.*, "Fast restoration in SDN-based flexible optical networks," in *Optical Fiber Communications Conference and Exhibition (OFC)*, March 2014, pp. 1–3.

[135] A. Giorgetti, F. Paolucci, F. Cugini and P. Castoldi, "Dynamic restoration with GMPLS and SDN control plane in elastic optical networks [Invited]," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 7, no. 2, pp. A174–A182, February 2015.

[136] H. Yang *et al.*, "Multipath protection for data center services in OpenFlow-based software defined elastic optical networks," *Optical Fiber Technology*, vol. 23, pp. 108 – 115, 2015.

[137] S. Savas *et al.*, "Backup reprovisioning with partial protection for disaster-survivable software-defined optical networks," *Photonic Network Communications*, pp. 1–10, 2015.

[138] C. Filsfils *et al.*, "Segment Routing Architecture," Internet Engineering Task Force, Internet-Draft draft-ietf-spring-segment-routing-07, Dec. 2015, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-ietf-spring-segment-routing-07

[139] A. Sgambelluri *et al.*, "First Demonstration of SDN-based Segment Routing in Multi-layer Networks," in *OFC*. Optical Society of America, 2015, p. Th1A.5.

[140] A. Sgambelluri *et al.*, "Experimental demonstration of multi-domain segment routing," in *ECOC*, Sept 2015, pp. 1–3.

[141] N. kukreja *et al.*, "Demonstration Of SDN-Based Orchestration For Multi-Domain Segment Routing Networks," in *ICTON*, Jul 2016.

[142] B. Lantz, B. Heller, and N. McKeown, "A Network in a Laptop: Rapid Prototyping for Software-defined Networks," in *ACM SIGCOMM Workshop on Hot Topics in Networks*, ser. Hotnets-IX. ACM, 2010, pp. 19:1–19:6.

[143] N. Handigol *et al.*, "Reproducible network experiments using container-based emulation," in *Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies*, ser. CoNEXT '12. ACM, 2012, pp. 253–264.

[144] S. Azodolmolky *et al.*, "SONEP: A Software-Defined optical Network emulation platform," in *International Conference on Optical Network Design and Modeling*, May 2014, pp. 216–221.

[145] G. Parulkar, T. Tofigh, and M. D. Leenheer, "Sdn control of packet over optical networks," in *Optical Fiber Communications Conference and Exhibition (OFC)*, Mar 2015, pp. 1–27.

[146] (2016) Infoblox. [Online]. Available: https://www.infoblox.com/resources

[147] (2016) LINC-OE: LINC-Switch for optical emulation. [Online]. Available: https://github.com/FlowForwarding/LINC-Switch/blob/master/docs/optical_extension.md

[148] (2016) Netphony GMPLS Emulator Repository. [Online]. Available: https://github.com/telefonicaid/netphony-gmpls-emulator

[149] R. Enns *et al.*, "Network Configuration Protocol (NETCONF)," RFC 6241, Oct. 2015. [Online]. Available: https://rfc-editor.org/rfc/rfc6241.txt

[150] K. Shiomoto *et al.*, "Requirements for GMPLS-Based Multi-Region and Multi-Layer Networks (MRN/MLN)," RFC 5212, Oct. 2015. [Online]. Available: https://rfc-editor.org/rfc/rfc5212.txt

[151] A. Ayyangar *et al.*, "Path Computation Element (PCE) Communication Protocol (PCEP)," RFC 5440, Oct. 2015. [Online]. Available: https://rfc-editor.org/rfc/rfc5440.txt

[152] R. Alimi, Y. Yang, and R. Penno, "Application-Layer Traffic Optimization (ALTO) Protocol," RFC 7285, Oct. 2015. [Online]. Available: https://rfc-editor.org/rfc/rfc7285.txt

[153] J. Medved *et al.*, "PCEP Extensions for Stateful PCE," Internet Engineering Task Force, Internet-Draft draft-ietf-pce-stateful-pce-14, Mar. 2016, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-ietf-pce-stateful-pce-14

[154] Open Networking Foundation. (2015, Nov) Common Information Model Overview. V1.1. [Online]. Available: https://www.opennetworking.org

[155] *Generic protocol-neutral information model for transport resources*, Recommendation ITU-T G.7711-Y.17022, ITU-T Std., Aug 2015.

[156] L. Lhotka, "JSON Encoding of Data Modeled with YANG," Internet Engineering Task Force, Internet-Draft draft-ietf-netmod-yang-json-10, Mar. 2016, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-ietf-netmod-yang-json-10

[157] ONF-OTWG, "Functional Requirements for Transport API," ONF Technical report: ONF TR-527, Jun 2016. [Online]. Available: https://www.opennetworking.org

[158] V. P. Beeram *et al.*, "YANG Data Model for TE Topologies," Internet Engineering Task Force, Internet-Draft draft-ietf-teas-yang-te-topo-04, Mar. 2016, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-ietf-teas-yang-te-topo-04

[159] T. Saad *et al.*, "A YANG Data Model for Traffic Engineering Tunnels and Interfaces," Internet Engineering Task Force, Internet-Draft draft-ietf-teas-yang-te-03, Mar. 2016, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-ietf-teas-yang-te-03

[160] Y. Lee *et al.*, "A Yang Data Model for ACTN VN Operation," Internet Engineering Task Force, Internet-Draft draft-lee-teas-actn-vn-yang-00, Jul. 2016, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-lee-teas-actn-vn-yang-00

[161] X. Zhang *et al.*, "YANG Models for the Northbound Interface of a Transport Network Controller: Requirements, Functions, and a List of YANG Models," Internet Engineering Task Force, Internet-Draft draft-zhang-ccamp-transport-ctrlnorth-yang-00, Mar. 2016, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-zhang-ccamp-transport-ctrlnorth-yang-00

[162] D. Dhody *et al.*, "A YANG Data Model for Path Computation Element Communications Protocol (PCEP)," Internet Engineering Task Force, Internet-Draft draft-pkd-pce-pcep-yang-05, Jan. 2016, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-pkd-pce-pcep-yang-05

[163] X. Zhang, B. Rao, and X. Liu, "A YANG Data Model for Layer 1 Network Topology," Internet Engineering Task Force, Internet-Draft draft-zhang-i2rs-l1-topo-yang-model-01, Oct. 2015, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-zhang-i2rs-l1-topo-yang-model-01

[164] Y. Lee *et al.*, "A Yang Data Model for WSON Optical Networks," Internet Engineering Task Force, Internet-Draft draft-ietf-ccamp-wson-yang-01, Apr. 2016, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-ietf-ccamp-wson-yang-01

[165] U. A. de Madrid *et al.*, "YANG data model for Flexi-Grid Optical Networks," Internet Engineering Task Force, Internet-Draft draft-vergara-ccamp-flexigrid-yang-02, Mar. 2016, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-vergara-ccamp-flexigrid-yang-02

[166] Open ROADM MSA. (2016, May) Open ROADM Overview. [Online]. Available: http://openroadm.org/download.html

[167] Open ROADM MSA. (2016, Jun) ROADM Network Model and Device Model. [Online]. Available: http://openroadm.org/download.html

[168] (2015) Standardization sector of International Telecommunication Union (ITU-T). [Online]. Available: http://www.itu.int/ITU-T/

[169] J. Medved *et al.*, "North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP," RFC 7752, mar 2016. [Online]. Available: https://rfc-editor.org/rfc/rfc7752.txt

[170] A. Doria *et al.*, "Forwarding and Control Element Separation (ForCES) Protocol Specification," RFC 5810, Oct. 2015. [Online]. Available: https://rfc-editor.org/rfc/rfc5810.txt

[171] T. Nadeau *et al.*, "An Architecture for the Interface to the Routing System," Internet Engineering Task Force, Internet-Draft draft-ietf-i2rs-architecture-15, Apr. 2016, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-ietf-i2rs-architecture-15

[172] *SDN standardization activity roadmap*, ITU-T JCA-SDN-D-001 Rev.2, ITU-T Std., July 2015. [Online]. Available: http://www.itu.int/en/ITU-T/jca/sdn/

[173] J. Schönwälder, "Overview of the 2002 IAB Network Management Workshop," RFC 3535, Mar. 2013. [Online]. Available: https://rfc-editor.org/rfc/rfc3535.txt

[174] *Architecture for the automatically switched optical network (ASON)*, Recommendation ITU-T G.8080/Y.1304, ITU-T Std., Feb 2012.

[175] (2016) Open Source SDN. [Online]. Available: http://opensourcesdn.org/

[176] H.-K. Lam *et al.*, "Usage of IM for network topology to support TE Topology YANG Module Development," Internet Engineering Task Force, Internet-Draft draft-lam-teas-usage-info-model-net-topology-03, May 2016, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-lam-teas-usage-info-model-net-topology-03

[177] M. Betts *et al.*, "Framework for Deriving Interface Data Schema from UML Information Models," Internet Engineering Task Force, Internet-Draft draft-betts-netmod-framework-data-schema-uml-03, Mar. 2016, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-betts-netmod-framework-data-schema-uml-03

[178] (2016) Open ROADM MSA. [Online]. Available: http://openroadm.org/home.html

[179] (2014) Open Networking Lab (ON.Lab). [Online]. Available: http://onlab.us/

[180] K. Bogineni et al, "Introducing ONOS - a SDN network operating system for Service Providers," Whitepaper, 2014. [Online]. Available: http://onosproject.org/wp-content/uploads/2014/11/Whitepaper-ONOS-final.pdf

[181] K. Teemu *et al.*, "Onix: A Distributed Control Platform for Large-scale Production Networks." in *OSDI*, vol. 10, 2010, pp. 1–6.

[182] P. Berde *et al.*, "ONOS: Towards an Open, Distributed SDN OS," in *Proceedings of the Third Workshop on Hot Topics in Software Defined Networking*, ser. HotSDN '14, 2014, pp. 1–6.

[183] (2016) OpenConfig working Group. [Online]. Available: http://www.openconfig.net/

[184] (2016) CORD: Central Office Re-architected as a Datacenter. [Online]. Available: ttp://opencord.org/about/

[185] (2016) Ciena's Blue Planet division. [Online]. Available: http://www.blueplanet.com/

[186] (2016) Juniper Networks. [Online]. Available: http://www.juniper.net/

[187] (2016) Sedona Systems. [Online]. Available: http://sedonasys.com/

[188] (2016) Calient Technologies. [Online]. Available: http://www.calient.net/

[189] (2016) Open ROADM MSA. [Online]. Available: http://www.openroadm.org/

[190] (2016) Lumen Networks. [Online]. Available: http://www.lumenetworks.com/

[191] (2016) Lumentum. [Online]. Available: https://www.lumentum.com

[192] (2016) Polatis. [Online]. Available: http://www.polatis.com/index.asp

[193] (2016) Fujitsu. [Online]. Available: http://www.fujitsu.com/

[194] R. Veisllari *et al.*, "Scalability analysis of SDN-controlled optical ring MAN with hybrid traffic," in *EEE International Conference on Communications (ICC)*, June 2014, pp. 3283–3288.

[195] K. Phemius, M. Bouet, and J. Leguay, "Disco: Distributed multi-domain sdn controllers," in *IEEE Network Operations and Management Symposium (NOMS)*, May 2014, pp. 1–4.

[196] D. Ongaro and J. Ousterhout, "In search of an understandable consensus algorithm," in *USENIX Annual Technical Conference*, 2014, pp. 305–320.

[197] H. Yin *et al.*, "SDNi: A Message Exchange Protocol for Software Defined Networks (SDNS) across Multiple Domains," Internet Engineering Task Force, Internet-Draft draft-yin-sdn-sdni-00, Dec. 2012, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-yin-sdn-sdni-00

[198] F. Benamrane, F. J. Ros, and M. B. Mamoun, "Synchronisation cost of multi-controller deployments in software-defined networks," *International Journal of High Performance Computing and Networking*, vol. 9, no. 4, pp. 291–298, 2016.

[199] L. Schiff, S. Schmid, and P. Kuznetsov, "In-Band Synchronization for Distributed SDN Control Planes," *ACM SIGCOMM Computer Communication Review*, vol. 46, no. 1, pp. 37–43, 2016.

[200] H. Ding *et al.*, "Experimental demonstration and assessment of multi-domain sdtn orchestration based on northbound api," in *Asia Communications and Photonics Conference*. Optical Society of America, 2015, p. ASu4F.1.

[201] S. Scott-Hayward, G. O'Callaghan, and S. Sezer, "Sdn security: A survey," in *IEEE SDN for Future Networks and Services (SDN4FNS)*, Nov 2013, pp. 1–7.

[202] C. Liou, "Is there a role for gmpls in transport sdn?" In TIP, Jan 2013.

[203] Cisco Visual Networking Index, "Global mobile data traffic forecast update, 2015-2020," 2016. [Online]. Available: http://www.cisco.com/

[204] J. Yuan, Y. Zheng, and X. Xie, "Discovering Regions of Different Functions in a City Using Human Mobility and POIs," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2012, pp. 186–194.

[205] Y. Xu *et al.*, "Affinity-based Human Mobility Pattern for Improved Region Function Discovering," *The Journal of China Universities of Posts and Telecommunications*, vol. 23, no. 1, pp. 60–67, 2016.

[206] C. Peng *et al.*, "Traffic-driven Power Saving in Operational 3G Cellular Networks," in *International Conference on Mobile Computing and Networking*, 2011, pp. 121–132.

[207] L. Budzisz *et al.*, "Dynamic Resource Provisioning for Energy Efficiency in Wireless Access Networks: A Survey and an Outlook," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 4, pp. 2259–2285, Fourthquarter 2014.

[208] R. Alvizu *et al.*, "Energy aware optimization of mobile metro-core network under predictable aggregated traffic patterns," in *IEEE International Conference on Communications (ICC)*, May 2016, pp. 1–7.

[209] G. M. Y. X. R. Alvizu, X. Zhao and A. Pattavina, "Energy efficient dynamic optical routing for mobile metro-core networks under tidal traffic patterns," *Accepted with Major revision for publication to IEEE/OSA Journal of Lightwave Technology*.

[210] M. Afanasyev *et al.*, "Analysis of a Mixed-use Urban Wifi Network: When Metropolitan Becomes Neapolitan," in *ACM SIGCOMM Conference on Internet Measurement*, 2008, pp. 85–98.

[211] R. Wang *et al.*, "Energy saving via dynamic wavelength sharing in twdm-pon," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 8, pp. 1566–1574, Aug 2014.

[212] Z. Zhong *et al.*, "Considerations of effective tidal traffic dispatching in software-defined metro ip over optical networks," in *Opto-Electronics and Communications Conference (OECC)*, June 2015, pp. 1–3.

[213] Z. Zhong *et al.*, "Energy efficiency and blocking reduction for tidal traffic via stateful grooming in ip-over-optical networks," *J. Opt. Commun. Netw.*, vol. 8, no. 3, pp. 175–189, Mar 2016.

[214] G. consortium, "GreenTouch Green Meter Research Study: Reducing the Net Energy Consumption in Communications Networks by up to 90 percent by 2020," A GreenTouch White Paper, 2013.

[215] F. Idzikowski *et al.*, "TREND in energy-aware adaptive routing solutions," *IEEE Commun. Mag.*, vol. 51, no. 11, pp. 94–104, November 2013.

[216] A. Ahmad *et al.*, "Energy-aware design of multilayer core networks [invited]," *J. Opt. Commun. Netw.*, vol. 5, no. 10, pp. A127–A143, Oct 2013.

[217] S. Tombaz *et al.*, "Impact of Backhauling Power Consumption on the Deployment of Heterogeneous Mobile Networks," in *IEEE GLOBECOM*, Dec 2011, pp. 1–5.

[218] H. Michael, "Using carrier Ethernet to backhaul LTE," Infonetics White Paper, 2011.

[219] R. Nadiv and N. Tzvika, "Wireless backhaul topologies: Analyzing backhaul topology strategies," Ceragon White Paper, 2010.

[220] M. Tornatore, G. Maier, and A. Pattavina, "Capacity versus availability trade-offs for availability-based routing," *J. Opt. Netw.*, vol. 5, no. 11, pp. 858–869, Nov 2006.

[221] Z. Zhu *et al.*, "Characterizing Data Services in a 3G Network: Usage, Mobility and Access Issues," in *IEEE International Conference on Communications (ICC*, June 2011, pp. 1–6.

[222] J. Robson, "Guidelines for lte backhaul traffic estimation," NGMN White Paper, pp. 1–18, 2011.

[223] W. Van Heddeghem *et al.*, "Power consumption modeling in optical multilayer networks," *Photonic Network Communications*, vol. 24, no. 2, pp. 86–102, 2012.

[224] M. Lippi, M. Bertini, and P. Frasconi, "Short-Term Traffic Flow Forecasting: An Experimental Comparison of Time-Series Analysis and Supervised Learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 2, pp. 871–882, June 2013.

[225] C. Chen and S. Banerjee, "A new model for optimal routing and wavelength assignment in wavelength division multiplexed optical networks," in *INFOCOM*, vol. 1, Mar 1996, pp. 164–171 vol.1.

[226] R. Bhandari, *Survivable Networks: Algorithms for Diverse Routing.* Norwell, MA, USA: Kluwer Academic Publishers, 1998.

[227] Simpy, "Event discrete simulation for Python," 2016. [Online]. Available: http://simpy.readthedocs.org/

[228] A. Akella *et al.*, "A measurement-based analysis of multihoming," in *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, ser. SIGCOMM '03, 2003, pp. 353–364.

[229] J. Chen, S. H. G. Chan, and V. O. K. Li, "Multipath routing for video delivery over bandwidth-limited networks," *IEEE J-SAC*, vol. 22, no. 10, pp. 1920–1932, Dec 2004.

[230] H. Sheng, C. Martel, and B. Mukherjee, "Survivable multipath provisioning with differential delay constraint in telecom mesh networks," *IEEE/ACM Trans. Netw.*, vol. 19, no. 3, pp. 657–669, June 2011.

[231] R. Muñoz *et al.*, "Design and experimental evaluation of dynamic inverse-multiplexing provisioning in GMPLS-controlled flexi-grid DWDM networks with sliceable OTN BVTs," in *ECOC*, Sept 2013, pp. 1–3.

[232] A. Ford *et al.*, "TCP Extensions for Multipath Operation with Multiple Addresses," RFC 6824, Oct. 2015. [Online]. Available: https://rfc-editor.org/rfc/rfc6824.txt

[233] M. Coudron *et al.*, "Cross-layer cooperation to boost multipath TCP performance in cloud networks," in *IEEE 2nd Int. Conf. on Cloud Networking (CloudNet)*, Nov. 2013, pp. 58–66.

[234] R. Muñoz, *et al.*, "Dynamic Differential Delay Aware RMSA for Elastic Multipath Provisioning in GMPLS Flexi-grid DWDM Networks," in *OFC*, 2014, p. W3A.2.

[235] F. Hu, Q. Hao, and K. Bao, "A survey on software-defined network and openflow: From concept to implementation," *Commun. Surveys Tuts.*, vol. 16, no. 4, pp. 2181–2206, Fourthquarter 2014.

[236] A. Tomaszewski and M. Pióro, "Minimizing label usage in MPLS networks," in *Asilomar Conf. Signals, Syst. and Comput.*, Nov. 2009, pp. 58–66.

[237] J. Santos *et al.*, "Optimized routing and buffer design for optical transport networks based on virtual concatenation," *IEEE J. Opt. Commun. Netw.*, vol. 3, no. 9, pp. 725–738, Sept. 2011.

[238] G. Appenzeller, I. Keslassy, and N. McKeown, "Sizing router buffers," *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 4, pp. 281–292, Aug. 2004.

[239] R. Alvizu, J. Valencia, and G. Maier, "Multipath optical routing with compact fiber delay line-based differential delay compensation," in *European Conference on Networks and Optical Communications (NOC)*, June 2016, pp. 58–63.

[240] J. Santos, *et al.*, "Optical transport network design with collocated regeneration and differential delay compensation," in *IEEE HPSR*, June 2012, pp. 204–209.

[241] Rack mount fiber optic delay lines. [Online]. Available: http://www.fiberplus.com/dloption8.html

[242] W. van Heddeghem, *et al.*, "Power consumption modeling in optical multilayer networks," *Photonic Network Communications*, vol. 24, no. 2, pp. 8–102, 2012.

[243] R. Bhandari, *Survivable Networks: Algorithms for Diverse Routing.* Norwell, MA, USA: Kluwer Academic Publishers, 1998.

[244] A. Betker et al., "Reference transport network scenarios," MultiTeraNet Report, July 2013.

[245] S. Orlowski *et al.*, "Sndlib 1.0—survivable network design library," *Networks*, vol. 55, no. 3, pp. 276–286, 2010.

[246] C. Filsfils *et al.*, "Segment Routing Architecture," Internet Engineering Task Force, Internet-Draft draft-ietf-spring-segment-routing-09, Jul. 2016, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-ietf-spring-segment-routing-09

[247] C. Filsfils *et al.*, "Segment Routing with MPLS data plane," Internet Engineering Task Force, Internet-Draft draft-ietf-spring-segment-routing-mpls-05, Jul. 2016, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-ietf-spring-segment-routing-mpls-05

[248] S. Sivabalan *et al.*, "PCEP Extensions for Segment Routing," Internet Engineering Task Force, Internet-Draft draft-ietf-pce-segment-routing-07, Mar. 2016, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-ietf-pce-segment-routing-07

[249] ONF Architecture Framework Working Group. (2014, Jun) SDN Architecture. SDN ARCH 1.0. [Online]. Available: https://www.opennetworking.org

[250] S. Bidkar *et al.*, "Scalable segment routing-a new paradigm for efficient service provider networking using carrier ethernet advances," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 7, no. 5, pp. 445–460, May 2015.

[251] L. Davoli *et al.*, "Traffic engineering with segment routing: Sdn-based architectural design and open source implementation," in *European Workshop on Software Defined Networks*, Sept 2015, pp. 111–112.

[252] A. Sgambelluri *et al.*, "Sdn and pce implementations for segment routing," in *Networks and Optical Communications - (NOC), 2015 20th European Conference on*, June 2015, pp. 1–4.

[253] F. Lazzeri *et al.*, "Efficient label encoding in segment-routing enabled optical networks," in *International Conference on Optical Network Design and Modeling (ONDM)*, May 2015, pp. 34–38.

[254] M. Coudron *et al.*, "Boosting cloud communications through a crosslayer multipath protocol architecture," in *IEEE SDN for Future Networks and Services (SDN4FNS)*, Nov 2013, pp. 1–8.

[255] C. Raiciu *et al.*, "Improving datacenter performance and robustness with multipath tcp," *SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 4, pp. 266–277, Aug. 2011.

[256] L. Magalhaes and R. Kravets, "Transport level mechanisms for bandwidth aggregation on mobile hosts," in *International Conference on Network Protocols*, Nov 2001, pp. 165–171.

[257] C. Raiciu *et al.*, "How hard can it be? designing and implementing a deployable multipath tcp," in *Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation*, ser. NSDI'12. Berkeley, CA, USA: USENIX Association, 2012, pp. 29–29.

[258] C. Paasch, R. Khalili, and O. Bonaventure, "On the benefits of applying experimental design to improve multipath tcp," in *Proceedings of the Ninth ACM Conference on Emerging Networking Experiments and Technologies*, ser. CoNEXT '13. New York, NY, USA: ACM, 2013, pp. 393–398.

[259] S. Savage *et al.*, "The end-to-end effects of internet path selection," *SIGCOMM Comput. Commun. Rev.*, vol. 29, no. 4, pp. 289–299, Aug. 1999.

[260] P. Key, L. Massoulie, and D. Towsley, "Path selection and multipath congestion control," in *IEEE International Conference on Computer Communications (INFOCOM)*, May 2007, pp. 143–151.

[261] M. Scharf and S. Kiesel, "Nxg03-5: Head-of-line blocking in tcp and sctp: Analysis and measurements," in *IEEE Globecom*, Nov 2006, pp. 1–5.

[262] S. Ferlin-Oliveira, T. Dreibholz, and O. Alay, "Tackling the challenge of bufferbloat in multi-path transport over heterogeneous wireless networks," in *2014 IEEE 22nd International Symposium of Quality of Service (IWQoS)*, May 2014, pp. 123–128.

[263] C. Paasch *et al.*, "Experimental evaluation of multipath tcp schedulers," in *Proceedings of the 2014 ACM SIGCOMM Workshop on Capacity Sharing Workshop*, ser. CSWS '14. ACM, 2014, pp. 27–32.

[264] R. Khalili *et al.*, "Mptcp is not pareto-optimal: Performance issues and a possible solution," *IEEE/ACM Transactions on Networking*, vol. 21, no. 5, pp. 1651–1665, Oct 2013.

[265] D. Wischik *et al.*, "Design, implementation and evaluation of congestion control for multipath tcp." in *NSDI*, vol. 11, 2011, pp. 8–8.

[266] Y.-C. Chen *et al.*, "A measurement-based study of multipath tcp performance over wireless networks," in *Proceedings of the 2013 Conference on Internet Measurement Conference*, ser. IMC '13. ACM, 2013, pp. 455–468.

[267] C. Paasch *et al.*, "Exploring mobile/wifi handover with multipath tcp," in *Proceedings of the 2012 ACM SIGCOMM Workshop on Cellular Networks: Operations, Challenges, and Future Design*, ser. CellNet '12. New York, NY, USA: ACM, 2012, pp. 31–36.

[268] B. Sonkoly *et al.*, "Sdn based testbeds for evaluating and promoting multipath tcp," in *IEEE International Conference on Communications (ICC)*, June 2014, pp. 3044–3050.

[269] S. Huang, C. U. Martel, and B. Mukherjee, "Survivable multipath provisioning with differential delay constraint in telecom mesh networks," *IEEE/ACM Transactions on Networking*, vol. 19, no. 3, pp. 657–669, June 2011.

[270] (2015) Linux SYSCTL. [Online]. Available: http://linux.die.net/man/8/sysctl

[271] (2015) Linux Secure Copy. [Online]. Available: http://linux.die.net/man/1/scp

[272] (2015) IPERF. [Online]. Available: https://iperf.fr/