Open-Source Network Optimization Software in the Open SDN/NFV Transport Ecosystem

Miquel Garrich, *Member, IEEE*, Francisco-Javier Moreno-Muro, *Student Member, IEEE*, María-Victoria Bueno Delgado, *Member, IEEE*, and Pablo Pavón Mariño, *Member, IEEE*

(Invited Tutorial)

***Abstract*—Transport ecosystems that combine software defined networking (SDN) and network function virtualization (NFV) are characterized by an unprecedented network control and resource dynamicity. Manual optimization is unmanageable. In this context, open systems that manage and orchestrate SDN/NFV-enabled networks offer programming frameworks that abstract the low-level particularities in the data-plane forwarding devices and in the hardware appliances that provide the IT resources. Although these open systems present notable complexity, their programming abstractions promote a client layer where third-party applications can provide different functionalities thus enabling optimization-as-a-service business opportunities. In this paper, we cover open-source optimization software initiatives for offline planning and online provisioning and orchestration of SDN/NFV networks. With this goal in mind, we first focus on open software (and framework) initiatives through a set of realistic use cases that require optimization in multi-layer optical transport scenarios and ecosystems that combine transport with IT resources. The importance of a joint optimization of both network and IT domains is emphasized, a new paradigm triggered by SDN/NFV technologies. We discuss the theoretical limits to algorithm performances, and review available open-source frameworks for problem modeling that enable the interaction with solvers. Finally, we focus on the Net2Plan open-source network planning tool, a Java-based software that suitably embraces the multiple features required in the optimization of joint transport network and IT resource SDN/NFV ecosystems. Recent works based on Net2Plan are reviewed to illustrate its suitability for rapid algorithm prototyping, and for interaction with SDN/NFV-enabled networks.**

***Index Terms*—Network function virtualization, open source software, optimization, software defined networking, transport networks.**

Manuscript received July 11, 2018; revised August 22, 2018; accepted August 23, 2018. Date of publication September 10, 2018; date of current version February 1, 2019. This work was supported in part by Spanish Government for the ONOFRE-2 project under Grant TEC2017-84423-C3-2-P, and in part by the European Commission for the H2020-ICT-2016-2 METRO-HAUL project under Grant G. A. 761727 and in part by the H2020-MSCA-IF-2016 INSPIRING-SNI project under Grant G. A. 750611. This paper was presented in part at the Optical Fiber Communication Conference and Exhibition, San Diego, CA, USA, March 2018. *(Corresponding author: Miquel Garrich.)*

M. Garrich and F.-J. Moreno-Muro are with the Universidad Politécnica de Cartagena, Cartagena 30202, Spain (e-mail: miquel.garrich@upct.es; javier.moreno@upct.es).

M.-V. Bueno Delgado and P. Pavón Mariño are with the Universidad Politécnica de Cartagena, Cartagena 30202, Spain, and also with E-lighthouse Network Solutions, Cartagena 30202, Spain (e-mail: mvictoria.bueno@upct.es; pablo.pavon@upct.es).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JLT.2018.2869242

I. Introduction

TELECOM operators' infrastructure is undergoing through major challenges due to the need to support the forecasted traffic growth which also presents high levels of dynamicity and a wide variety of requirements [2]. Indeed, Cisco predictions announce a 46% compound annual growth rate of mobile data traffic and 24% of global IP traffic. This impacts on the infrastructure of carriers and ISPs given by the diversity on the traffic needs among machine-to-machine (M2M), Internet of Things (IoT), Live Internet video, consumer video on demand, virtual and augmented reality, among others [2]. Additionally, the presence of multiple domains, numerous technologies and equipment acquired from multiple vendors, poses notable operational complexity every time carriers need to provision new demands or services, or implement new business models. On top of that, key performance indicators of the forthcoming 5 G era drive implications at an operational level while enforcing the usage of "*a scalable management framework enabling fast deployment of novel applications, including sensor based applications, with reduction of the network management OpEx by at least 20% compared to today*" [3]. Consequently, carriers' infrastructure characterized by quasi-static IP-over-optical (wavelength division multiplexed, WDM) networks shall undergo through fundamental changes in the near future.

Software-Defined Networking (SDN) and Network Function Virtualization (NFV) are two key technologies to accomplish the above referred mission. SDN is based on decoupling the forwarding data-plane actions from the control-plane logical decisions [4]. This principle enables flexible forwarding and steering of traffic because network switches become simple forwarding devices and the control logic is implemented in a logically centralized controller [5]. NFV is based on implementing network functions traditionally based on bare metal hardware, as software pieces running in commodity servers. Such *virtualized network functions* (VNFs) can then be rapidly and flexibly instantiated, scaled up and down, in the network carrier data centers [6], [7]. SDN and NFV are complementary technologies that enable high levels of network programmability as flexible placement of VNFs can be jointly exploited with centralized traffic flow policies and (re)configurations. Indeed, the combination of SDN and NFV enables unprecedented levels of network control and dynamicity.

Automated optimization procedures are essential to efficiently employ the resources available in SDN/NFV-enabled transport networks: operation complexity is unmanageable with traditional manual approaches. In this context, the SDN/NFV transport is witnessing the appearance of numerous initiatives that provide open systems to control, manage and orchestrate SDN/NFV-enabled networks [8]–[11]. These open systems are characterized by programming frameworks that abstract the low-level particularities in the data-plane forwarding devices and in the hardware appliances that provide the IT resources. It is worthwhile mentioning that notable differences exist in the abstractions of the physical characteristics in the case of transport devices (e.g., optical and electronic packet layers) when compared to IT resources. Indeed, the fundamental difference between transport (SDN) [5], [9] and IT resources (NFV) [6], [8], [10] clearly implies a different set of features to be abstracted in each open system, which also depends on network types, specific software architectures (e.g., control plane implementations) and management strategies. Each of these open initiatives address specific needs ranging from common application programming interfaces (APIs) definition for control and management of network systems [12]–[15] to resource orchestration in multi-tenant infrastructures [10], [16]. Most of these works rely on OpenStack [8] for managing the IT infrastructure. OpenStack is an open-source private cloud platform that controls, through a graphical user interface (GUI), command line interface (CLI) or open APIs, large pools of computing, storage, and networking resources in data center scenarios.

In this paper, we cover open-source optimization software initiatives for offline planning and online provisioning and orchestration of SDN/NFV optical transport networks. We illustrate a set of realistic use-cases clearly benefiting from automatic optimization. Firstly, we begin with the specific needs of the optical transport use case where open systems [12]–[15] enable different levels of disaggregation, and provide the means to perform classical physical-layer optimization e.g., power control, via third party tools. Secondly, we illustrate the use-case of network resource optimization in multi-layer (e.g., IP over WDM) transport scenarios in which a SDN-controller promotes Optimization-as-a-Service (OaaS). Thirdly, we target NFV use-cases in a virtual infrastructure manager (VIM) domain, e.g., OpenStack, where VNF instances in a service function chain (SFC) can be flexibly scaled up and down, to dynamically adapt the consumed IT resources to the traffic load. Finally, a use case will be discussed in which network SFCs across multiple VIM instances controlling multiple data centers, are connected in an optical transport network requiring optimization to jointly allocate network and computing resources.

To address the algorithm challenges, we first discuss the theoretical limits to algorithm performances since most optimization problems in the discussed use cases are proven to be inapproximable. Then we review practical aspects in choosing among the available algorithmic approaches, mainly *solver*-based formulations and heuristics. *Solvers* are specialized libraries devoted to find numerical solutions to structured optimization problems in the form of linear or nonlinear constrained formulations. A hybrid approach so-called math-heuristics is covered, where the problem is divided into "subproblems/subroutines" that exploit multiple interactions with the solver, e.g., configured with a maximum running time. In this context, we overview (i) different commercial and open-source solvers available, and (ii) software modelling alternatives, that ease the interaction with them. We discuss the key performance aspects in the solver-modeler interaction including the parametrizing effort required in the problem formulation.

Finally, we focus on open-source network optimization suites and suitable initiatives for the SDN/NFV context. We concentrate on Net2Plan [11], an open-source Java-based software tool, which suitably embraces multiple features required in the optimization of SDN/NFV networks. Net2Plan capabilities for rapid algorithm prototyping are highlighted, also interacting with a wide range of commercial or open-source *solvers* [17], using a provided modeling library JOM (Java Optimization Modeler). Net2Plan merits are discussed to address the set of realistic previously referred use cases, and we deepen in the need to perform the optimization of the NFV resource allocation jointly with the mutli-layer transport resources decision.

The remainder of the paper is organized as follows. Section II presents a set of use cases that require optimization to address practical problems in open software initiatives. Section III discusses algorithmic aspects, including the open-source options available for algorithm development. Section IV reviews open-source software alternatives to interact with solvers. In Section V, we deepen into the characteristics and functionalities of Net2Plan and how it implements the optimization of the practical problems reported in Section II. Section VI summarizes the main aspects addressed in the paper.

II. Use Cases: SDN/NFV Optimization in Open Systems

This section divides the open systems that control, manage and orchestrate SDN/NFV-enabled networks in four major areas that are currently attracting the interest in the community. First, we cover numerous initiatives in the context of so-called optical disaggregated systems. Then, we address the SDN-control of network resources in multi-layer transport scenarios. Subsequently, we overview open initiatives that address the management of IT resources, i.e., NFV infrastructure in data center environments. Finally, we focus on the open frameworks to jointly control and manage IT and network resources in optical transport networks. We discuss a use case in each of these areas in which open-source software is already available to perform optimization procedures in realistic scenarios, i.e., beyond traditional theoretical/abstract academic problems.

A. *Optical-Layer Transport Control*

Traditional transport infrastructure of telecom operators is characterized by a vertical integration, which commonly means that a vendor deploys network equipment jointly with the control plane functionalities, a network management system and also provides to the carrier a proprietary software tool to perform optimization procedures. This approach creates the so called "vendor islands" in which nation-wide carriers face numerous issues (e.g., changeability, interoperability or compatibility) when performing, for instance, end-to-end resource allocation in their multi-vendor infrastructure. Additionally, such vendor-locked vertical integration reduces flexibility and limits innovation and evolution of the network infrastructure. This scenario motivates the use case:

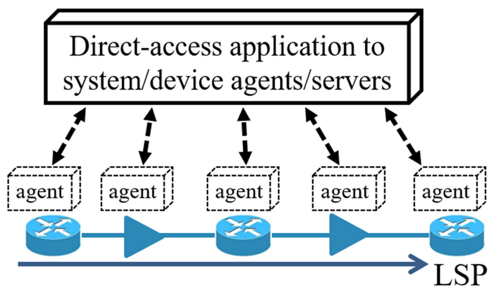


Fig. 1. Use case A: LSP quality of transmission optimization configuring: transponder transmission power, amplifier gain, WSS filtering.

1) Lightpath Quality of Transmission Optimization in a Disaggregated Network: as depicted in Fig. 1. In this use case the input is composed of a set of optical connections, their signal quality requirements (e.g., OSNR, polarization and chromatic dispersion tolerances) and the physical-layer network parameters. The problem outputs are the device and components configurations (transmission power, amplifier gain, WSS filtering). In partially disaggregated networks, this use case is addressed by a potentially third party SDN controller that interacts with the optical equipment via its south-bound interface (SBI) to receive monitor information and enforce configuration control orders. Then, third party power optimization or physical layer optimization applications at the north bound interface (NBI) of the SDN controller could interact with multivendor transport infrastructures.

In this context, initiatives that target white-box and openness for the SBI of SDN-controlled optical networks are attracting the interest of telecom operators because their ultimate goal is to define an open unified vendor-independent API [18]. These initiatives rely on two major components. On the one hand, the *NETCONF protocol* [19] created by the IETF, a remote procedure call (RPC) protocol that uses extensible markup language (XML) messages to perform changes in both static and runtime configurations of network devices. NETCONF establishes main operations for configuration and state management aside from allowing extensions by means of defined capabilities [19]. On the other hand, the YANG [20] modeling language is used for rapid prototyping of APIs. YANG descriptions allows mapping configuration parameters and status information for each device in a tree-data structure, describing the information model. Then, full NETCONF-based APIs can be created without manual intervention, by software suites that automatically process the YANG description. In particular, NETCONF sessions are defined on a per-device basis however formalized using the YANG modelling language [20]. Further details on these aspects can be found in [21].

Academic initiatives to control optical components recently proposed open YANG models for elastic optical networks (EONs) [21], with focus on the monitoring functionalities [23] and specific models to manage sliceable transponders [24]. Optical network failure issues are addressed in [25]. More recently, Sgambeluri *et al.* demonstrated ROADM white-box machine learning capabilities including various levels of disaggregation, NETCONF/YANG control, telemetry and spectrum-based advanced monitoring functionalities [26]. In general terms, given the possibility to propose innovative research directions, academic initiatives commonly propose specific model extensions precisely to address these aspects. Subsequently, specific projects (e.g., Metro-Haul) and industry-lead open initiatives pursue its standardization. On the industrial side, a major goal is to implement the so called *open line systems* (OLS), commonly in the form of optical topologies that are vendor-interoperable both at the data and control planes. In the following, we overview the most relevant OLS-related initiatives which also pursue the openness and disaggregation of carrier-grade networks.

OpenROADM [12] is a Multi-Source Agreement (MSA) that defines optical interoperability specifications and YANG models which currently counts on 15 members including world leading vendors and telecom operators. OpenROADM main goal is to define multi-vendor, interchangeable and inter-workable optical functions with standard APIs written in YANG to be accessed from an SDN controller using NETCONF. YANG models include pluggable optics with interoperable line-sides and flexible interoperable ROADMs with colorless and directionless (CD) and contentionless (CDC) features.

The *OpenConfig* [13] project provides a common data model for management interfaces that network operators can use to configure and monitor any equipment regardless of the vendor. OpenConfig includes a set of vendor neutral data models that address different technologies, not only optical, and which are derived from operational needs, use cases and requirements from operators. In the optical domain, OpenConfig describes a set of five models for the optical transport systems.

OpenDevice [27] is an adaptation of the proposals available in OpenROADM and OpenConfig with the addition of a declarative part with specific functionalities. In particular, Infinera and Lumentum recently demonstrated [27] an automated service management and automated optical power control based on YANG modelling in which optical signals were expressed between the two vendors' ROADMs and terminals. OpenDevice is Lumentum's implementation on white-box for optical networks [28]. Notably, this industrial initiative is also followed by other equipment vendors that similarly target disaggregated and open systems. Clear examples are Fujitu's 1FINITY series which leverages on functionally disaggregated hardware components and re-aggregates their functionality through enhanced software [29], and Juniper's Programmable Photonic Layer OLS solution [30].

A recent initiative in this area is Facebook's collaborative Telecom Infrastructure Project (TIP) [31] aiming to drive innovation across the entire telecom landscape. Within TIP, the Open Optical Packet Transport (OOPT) project group [32] released Voyager [33], a white-box transponder and routing solution that adapts the Wedge top-of-rack switch with pluggable optical transceiver modules supporting up to 200 Gb/s. The OOPT group aims to define an open API architecture to make multi-vendor, multi-platform management integration easier [34]. Within the OOPT, the Physical Simulation Environment (PSE) group proposes a framework to estimate the quality of transmission (QoT) of one or more channels operating lightpaths over a given network route. The framework is named GNPY and is fully community-driven and available in an open source library [35]. GNPY relies on the Gaussian Noise (GN) model [36], and recently reported validation results in large-scale testbeds [37], [38]. In particular, [38] reports the GNPY validation in a mixed-fiber test-bed network at Microsoft labs in which commercial transceivers from eight different vendors operated in the C-band achieve propagation distances up to 1945 km.

The Open Networking Foundation (ONF) recently commenced the Open and Disaggregated Transport Network (ODTN) project [14] as an operator-led initiative to build a *reference implantation* for data center interconnections using disaggregated optical equipment, open and common standards, and open source software. ODTN's target is to enable a white-box optical 'peripherals' to be combined into complete solutions through three major aspects: (i) disaggregating DWDM systems (i.e., transponders and Open Line Systems, amplifiers, multiplexers, all-optical switches and ROADMs), (ii) providing an Open Network Operating System (ONOS)-based control and configuration of the DWDM system [9] and (iii) using open and common data models, APIs and protocols. At the moment of writing this paper, ODTN is in its phase 1 (of three), addressing the modeling and API definition of point-to-point systems following the OpenConfig principles for the SBI and Transport (T)API [15] for the NBI [39].

Although several of the referred open initiatives address problems in the same range and apparently overlap in some extent, differences exist between the proposed models and implementations. For instance, the three device-oriented proposals OpenROADM, OpenConfig and OpenDevice are capable to configure optical devices and components based on software agents compiled from their respective YANG models. Nonetheless, OpenROADM representations are capable to provide a network-wide representation suitable for path computation, whereas OpenConfig and OpenDevice do not provide this capability. For a detailed comparison between the proposed models, their differences and overlapping aspects, the reader may refer to [40]. Note that ODTN and TIP embrace a wider range of aspects from this use-case on lightpath quality of transmission optimization in a disaggregated network and the following one regarding network-wide control. Indeed, ODTN pursues a reference implementation with network-wide view and control based on the OpenConfig and TAPI combination whereas TIP currently considers the proposals from its subgroups for the white-box control (i.e., Voyager) and GNPY from the PSE subgroup for the network-wide optimization.

B. Control of Multi-Layer IP Over WDM Transport Networks

Transport networks with control plane architectures that exploit the global view of the network can take centralized decisions suitable for optimization, long-term planning and other network strategic aspects (e.g., policy definition) [41]. Indeed, the control and data plane separation enabled by SDN promotes a client layer above the SDN controller where third-party applications can offer different functionalities (e.g., optimization services). These applications commonly exploit proposed standards for the NBI of the SDN controllers such as the Transport (T)API [15]. The *path computation service* defined in TAPI is an opportunity for third-party optimization engines implementing it, to make network allocation decisions. This leads to the use case:

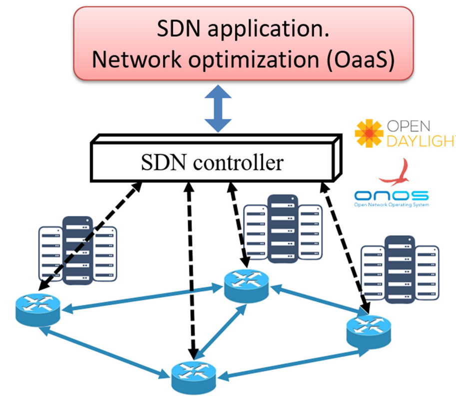


Fig. 2. Use case B: Optimization of network resources in SDN networks.

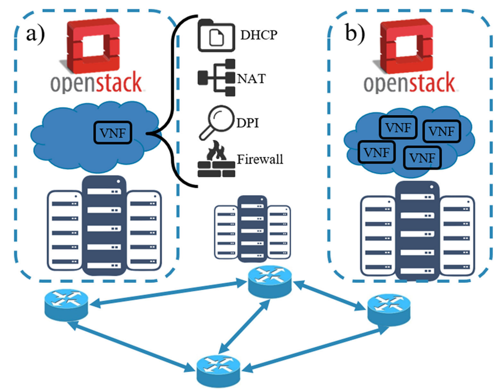


Fig. 3. Single OpenStack domain for (a) use case C.1: Optimization of a single VNF (VM) and (b) use case C.2: VNFs composing a service chain.

1) Optimization of Network Resources in SDN Networks: (see Fig. 2) which covers diverse sub-problems: label-switched path (LSP) computation and allocation, periodic re-optimization, capacity and long-term network planning. It receives as inputs LSP requests, network status retrieved from the SDN controller (e.g., traffic engineering data base –TED); generates as outputs the configuration parameters of LSPs and allocation of network resources, set of network configurations, or policies and strategic decisions; and is addressed with OaaS applications implemented on top of the SDN controller interfaced via NBI.

C. IT Resource Allocation

Infrastructure as a Service (IaaS) within the cloud computing paradigm refers to the technology in which traditional hardware resources are dynamically provided as virtualized entities and following the "as a service" approach. The baseline service is the instantiation, interconnection and life cycle management of virtual machine (VM) instances that make use of the data center computing services, storage facilities and communication service among VMs through the data center network. VIMs are in charge of providing these services being responsible of the creation/migration/deletion of VM instances, disk images, and the management of the VM network interfaces. OpenStack [8], an open-source option, is positioned as the most popular VIM.

1) Optimization of IT Resources for a Single VNF (VM) in a Single VIM Domain: as depicted in Fig. 3(a). This use case

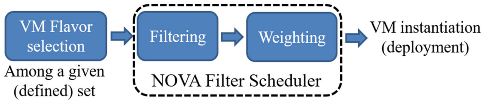


Fig. 4. Nova workflow for VM instantiation.

receives as input (i) the VM requirements in terms of CPU, HD, RAM & Enhanced Platform Awareness (EPA) information, and (ii) servers status (CPU, HD, RAM occupancy & EPA info); generating as output the VM allocation in the servers. EPA information consists of fine grained detailed descriptions of the hardware requisites and available resources like CPU&RAM interconnection topologies or PCI express direct connection availability. This information is relevant for instantiating VNFs with low-latency requirements.

Considering OpenStack [8] as the VIM to address this use case, its compute service project *Nova* provides open APIs for massively scalable access to compute resources, including bare metal, virtual machines, and containers [42]. Within Nova, a sequence of steps is followed prior to the VM instantiation as depicted in Fig. 4. *Flavors* define the compute, memory, and storage capacity of (Nova) computing instances. Users specify VM requisites as a flavor listing configurations for a virtual server (i.e., combinations of vRAM, vDisk, vCPUs). OpenStack provides procedures for tuning the VM allocation decisions in the available infrastructure. This is implemented through a so-called *Filter Scheduler* (also known as scheduler) module, which given a VM instantiation request, filters and ranks the list of hosts where it can be instantiated, i.e., a specific hardware appliance, in the list of acceptable hosts in the data center [43]. User-developed filters can be created and integrated into OpenStack. This opens the door for specialized external optimization software to optimize the data center usage. For instance, the processing load can be balanced among the hosts to optimize the power consumption (e.g., switching off servers when possible). Additionally, the scheduler allows the definition of availability zones and e.g., hosts connected to the same power supply can be identified with a given tag such that VNF placement can be tuned accordingly, to improve service failure tolerance. This functionality is permitted in the Nova (computing service) and Neutron (networking service) projects [44]. Furthermore, the scheduler provides control on the placement of VNFs with specific hardware considerations to improve performance in terms of latency and throughput. Examples are CPU pinning option that assigns a vCPU to a particular host CPU, and the (EPA) templates that permit defining Non-Uniform Memory Access (NUMA) topologies in which system memory is divided into cells or nodes that are associated with particular CPUs [45]. In this line, authors in [46] propose an extension of the OpenStack scheduler that enables a data center internal network-aware placement of instances by taking into account bandwidth constraints between nodes using an external resource tracker that monitors the bandwidth allocation.

2) Optimization of a Service Chain Composed of Multiple VNFs (VMs) in a Single VIM Domain: (see Fig. 3(b)) in which the VIM is requested to allocate a *network service* composed of a set of VNFs to be interconnected through the data center network. This use case considers as input the set of VNFs, virtual connections and metadata; and generates as output the allocation

TABLE I
OVERVIEW OF OPEN SOFTWARE/FRAMEWORK SOLUTIONS

Use case	Initiatives
A. Lightpath quality of transmission optimization in a disaggregated network.	OpenROADM [12] OpenConfig [13], ODTN (SBI) [14], OpenDevice [27], TIP (OLS) [31] [34]
B. Optimization of network resources in SDN networks.	T-API [15], TIP (C-API, PSE/GNPY) [31] [34], ODTN (NBI) [14]
C.1. Optimized allocation of IT resources to a single VM in a single VIM domain.	OpenStack [8] (Nova project [42])
C.2. Optimization of a service chain in a single VIM domain	OpenStack [8] (Heat project [48], Watcher project [50][42])
D. Network service allocation across multiple VIMs connected in a transport network	OSM [10], ONAP [16], OpenStack [8] (Tacker project [58][42])

of VMs in the servers and the configuration of the virtual links between them. In OpenStack, the *Heat* service orchestrates the allocation of network services, defined in *Heat Orchestration Template* (HOT) templates, in the underlying computing and networking infrastructure [47]. HOT templates are constructed in the form of text that can be treated like code and follow the structure: version, description, parameters, (stack of) resources, and output. Within Heat, an autoscaling group can be created to bundle together computing and network resources that can be allocated on demand for a VNF [48]. This functionality is compatible with the Amazon Web Services (AWS) cloud formation [49] and permits the so called horizontal autoscaling so that OpenStack can dynamically adjust the number of VMs to instantiate of a given type, to adapt to the traffic processed by the multi-VM VNF. OpenStack permits addressing this use case tuning several autoscaling thresholds to, for instance, avoid unproductive short-lived re-scalings.

This use case can also leverage on the *Watcher* project in OpenStack, a more recent and less mature module. Watcher targets a flexible and scalable resource optimization service including a metrics repository, optimization processor and an action plan applier [50]. Specifically, Watcher aims to ease a complete optimization of the OpenStack reading the monitoring information e.g., from the Ceilometer service (e.g., number of vCPUs, CPU's utilization, used memory) and permits to plug-in optimization algorithms using it. More details on the Watcher architecture can be found in [51]. Finally, Net2Plan-OpenStack module is an ongoing project for optimizing simple VMs and network service allocations in an OpenStack [52].

Table I provides an overview on the surveyed open software/framework initiatives covered in this section. As pointed out at the end of subsection II.A, differences exist between device-oriented YANG modeling proposals for the use-case A. Additionally, projects with a wider scope such as TIP and ODTN address both use-case A and B. Similarly, it is worth to mention that OpenStack services (i.e., subprojects) also cover use cases C1, C2 and a specific MANO implementation for use-case D.

D. Joint Transport Network and IT Resource Allocation

Carrier transport networks are evolving towards a distributed data center infrastructure to jointly exploit the benefits of SDN and NFV. Initiatives such as Central Office Re-architected as a Datacenter (CORD) [53] are attracting the interest of telecom

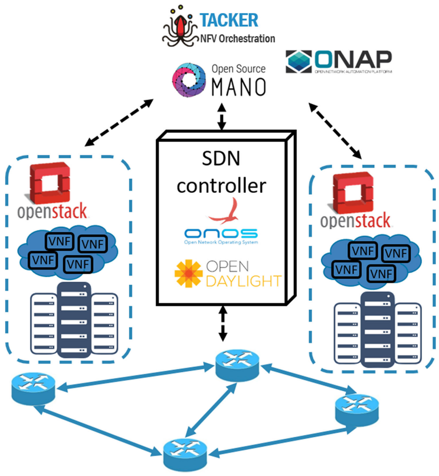


Fig. 5. Use case D: Network service across multiple OpenStacks connected in a network.

operators and promise benefits in terms of agility, efficiency and performance. Recent works show that placing functionalities and content closer to the end users may partially alleviate the impact in core/backbone networks [54]. These SDN/NFV transport ecosystems are characterized by the joint optimization of the optical transport network and the IT resources. In this context, the use case:

1) Network Service Allocation Across Multiple VIMs Connected in a Transport Network: (see Fig. 5) aims to provision a service chain across multiple-VIMs in which network and computing resources are *jointly* allocated. This use case receives as inputs the template with the set of VNFs (e.g., AWS-style, HOT), virtual connections and metadata; and generates as an output (i) in which data center and server each VNF is instantiated, and (ii) the configuration of the inter-data center flows. Note that decisions (i) and (ii) are naturally coupled, and should be addressed jointly. This requires having visibility of both the available resources in the data centers and in the optical transport network connecting them.

In the SDN/NFV context, two major open-source initiatives exist that include a global IT plus transport network view in its roadmap: ETSI Open Source NFV Management and Orchestration (MANO) [10] and Open Network Automated Platform (ONAP) [16].

MANO approach is a part of the ETSI NFV architecture, the module in charge of orchestrating the VNF placement and life cycle in one or more data centers, potentially connected by a transport network. Open Source MANO (OSM) [10] is an open-source MANO implementation under the umbrella of ETSI [55], which delivered its fourth release in May 2018. OSM permits the deployment of network services connecting VNFs in multiple data centers, and defines a so called WIM module (WAN Infrastructure Manager), as a software element abstracting and giving access to the transport network via open programmatic interfaces. A WIM implementation in the context of optical transport networks is being pursued by the H2020 Metro-Haul project [56], targeting its integration in OSM fifth release. The joint IT and optical transport network planning, and the optimized allocation of network services, relies in Metro-Haul project on Net2Plan tool, as a third party open-source software providing an OaaS. This has been demonstrated in the works [17] and [57], further detailed in Section IV.

Within OpenStack, the Tacker project provides a generic VNF Manager and Orchestrator based on ETSI MANO Architectural Framework and provides a functional stack to Orchestrate Network Services end-to-end using VNFs [58].

On the other hand, the ONAP initiative aims to provide an open real-time/policy-driven platform for management and orchestration of physical and virtual network functions so that network, IT and cloud providers can rapidly automate new services and support complete lifecycle management [16]. ONAP considers as examples VoLTE and residential vCPE to apply its vendor-agnostic service framework. ONAP commenced in March 2017 and is a founder member of the Linux Foundation Networking, a new entity that fosters collaboration and excellence across networking projects [59].

III. Addressing the Use Cases: Algorithms, Modelers and Solvers

In this section, we review the open-source tools that support the prototyping and production-grade development of the optimization algorithms that lay at the heart of the SDN/NFV network optimization tools. We first start with some basic insights on the theoretical limits of the performances of the algorithms, coming from Complexity Theory, and discuss how they affect in the selection of the algorithmic techniques to apply. Then we discuss three algorithmic techniques for facing *non-approximable* problems. In particular, we first discuss (i) pure heuristic approaches, then (ii) the utilization of problem *modelers* together with specialized numerical libraries called *solvers*, and (iii) math-heuristics, a combination of (i) and (ii). Open-source and free modeling libraries and solvers will be listed, as well as the major commercial alternatives.

A. Theoretical Limits to Algorithm Performances

A key outcome of Complexity theory is its classification of the mathematical problems, according to the performance limits to the algorithms solving them. In our context we are interested in *optimization problems*, which search for a solution of minimum cost or maximum profit, represented by a real number, that is feasible according to given constraints.

In rough words, we say that an optimization problem p belongs to complexity class NPO (non-deterministic polynomial time optimization problem), if given any tentative solution x to the problem, it is possible to compute its cost in polynomial time, and it is possible to check if it is feasible in polynomial time. NPO problems which can be optimally solved in polynomial time are said to be in class P. The difficulty among the rest of NPO problems is ranked according to the existence or not of algorithms that find ε-approximate solutions in polynomial time, that is, solutions with a cost at most ε-times worse than the optimal. The class APX is composed of those problems for which a polynomial approximation algorithm exists, for at least an ε value. The class ε-APX is composed of the APX problems for which there are ε-approximations, or better.

TABLE II
COMPLEXITY OF OPTIMIZATION PROBLEMS OF INTEREST IN NETWORK DESIGN [60]

Name	Description	Complexity
Conv	General convex programs	Polynomial (P)
ILP	General integer linear programs	NPO-complete
BLP	General binary linear programs	NPO-complete
Min-kMST	k-minimum cost spanning trees	Polynomial (P)
Min-kSP	k-minimum cost paths	Polynomial (P)
Min-Ste	Min cost multicast tree (Steiner tree)	0.55-approx., APX-complete
Max-Clique	Maximum size clique	NPO-complete
Min-TSP	Min cost ring	NPO-complete
Min-NonBif	Min congestion non-bifurcated routing	2.23-approx., APX-complete
Max-IntegralFlow	Max integral k multicommodity flow on trees	1-approx., APX-complete
Min-NodeLocation	Min cost node location, no connectivity limit	1.4-approx., APX-complete

Many problems are known to be non-approximable in polynomial time,[1] which in practice means that all polynomial approximation methods have guarantees that get worse without bound as the problem size increases. The NPO-complete class includes all these problems, the hardest NPO problems.

Table II brought from [60] lists the complexity of several optimization problems that are of interest in the area of network design. For instance, general unstructured Binary, Integer or Mixed-Integer Linear Programs (BLP, ILP, and MILP, respectively) are not approximable. This happens also to other problems like the maximum clique or the minimum cost ring, which appear as subproblems in common network instances.

The general bottomline is that practical network design problems are in general not approximable in polynomial time, and have to be addressed via algorithmic techniques that do not pursue apriori approximation guarantees.

B. Choosing the Right Optimization Technique

In general, the SDN paradigm promotes a centralized decision making, meaning that there is an entity in the network that collects and delivers to the optimization engine the necessary inputs for solving the problem, and applies the action decided by it. In this paper, we are assuming such approach.[2]

Choosing the proper technique to develop an optimization algorithm is crucial, and there is nothing like a one-size-fits-all. The decision of what technique to use first depends on the problem approximability limits. If the problem is known to be solvable or approximable in polynomial time, the best option is adapting a suitable algorithm from the literature. Experience dictates that it is unlikely that running times exceed few milliseconds for practical problem sizes in commodity hardware. This is non critical in this context, comparing to, for instance, the message exchange times in the software stack with the SDN controllers (milliseconds), software instantiation times of VNFs (tens of seconds, several minutes), or physical reconfiguration times in the optical transport (e.g., seconds/minutes in amplifier power control).

Three main options exist for addressing non-approximable problems: heuristics, formulation-solving via an external solver, or a combined approach.

On the one hand, *heuristic algorithms* are ad-hoc developed schemes for a specific purpose, as creative combinations of different so-called meta-heuristic techniques, e.g., genetic algorithms, tabu lists, swarm optimization, or greedy approaches, among others. Designing the meta-heuristic techniques and tuning their parameters to smartly explore the solution space is greatly problem-dependent, and requires practical experience [60].

On the other hand, the *formulation-solving approach* is based on (i) modeling the problem as a constrained optimization problem (typically Mixed Integer Linear Programs -MILP), and (ii) relying on specialized numerical libraries called *solvers* for solving them. Due to the above referred NPO-complete complexity, solvers are usually configured with a *maximum running time*, so the best solution found so far, if any, is returned after the given time limit is reached. For high performance executions, callback techniques can be employed to drive the solver progress through multiple internal iterations.

The formulation-solving approach has advantages. First, its simplicity, compared to heuristic design, since there is no need to tune heuristic parameters, typically an empirical and error prone process, which requires significant optimization skills. Second, its performance: *solvers* are tested libraries implementing highly specialized mathematical methods to explore the solution space in computationally efficient forms. As a disadvantage, large-scale problems are typically outside solver capabilities, although the frontier in network size is totally problem dependent. As an example, MILP *solvers* can solve in few seconds some practical routing and wavelength assignment instances with a few millions of constraints in conventional laptops [61].

Finally, *math-heuristic techniques* are an approach combining heuristics and optimization via *solvers*. In the general case, the problem is addressed by a heuristic algorithm, that iteratively drives the exploration into smaller pieces of the solution space, so each of such explorations involve formulating a subproblem and relying on a solver, with maximum computation time limit, for addressing it. The advantage of such method is exploiting the computational efficiency of the solvers, and enabling its utilization in large-scale problems.

C. Practical Aspects in Solver Interactions

Developing a network optimization algorithm that relies on a numerical solver involves a number of practical aspects to consider. Two software elements are involved, aside the main program (where the input parameters sit, and the output should be delivered). These are, (i) the problem *modeler*, and (ii) the *solver*.

1) Optimization Modelers: Optimization modelers or *modeling software* are external programs or libraries that receive a description of an optimization problem in a human readable syntax, and interface with the *solver* of choice to solve it. In MILP solvers, such interface is typically based on providing to the solver the matrices describing the linear constraints and objective function computed from the problem data, and

[1]This and other complexity limits have been proved under the hypothesis that P≠NP conjecture is true.

[2]This leaves aside distributed techniques where agents that loosely communicate each other cooperatively solve a problem (see [60]. for a rationale on this topic).

TABLE III
MODELERS

Name	Access	Type
AMPL [64]	COMMERCIAL	External
GAMS [65]	COMMERCIAL	External
AIMSS [66]	COMMERCIAL	External
LINGO [67]	COMMERCIAL	External
MPL [68]	COMMERCIAL	External
TOMLAB [69]	COMMERCIAL	External
CMPL [70]	FREE / OPEN-SOURCE	External
CVX [71]	FREE / OPEN-SOURCE	Library (MATLAB)
Pyomo [72]	FREE / OPEN-SOURCE	Library (Python)
PuLP [73]	FREE / OPEN-SOURCE	Library (Python)
CyLP [74]	FREE / OPEN-SOURCE	Library (Python)
yaposib [75]	FREE / OPEN-SOURCE	Library (Python)
FLOPC++ [76]	FREE / OPEN-SOURCE	Library (C++)
JOM [77]	FREE / OPEN-SOURCE	Libray (Java)

TABLE IV
SOLVERS

Name	Access
GUROBI [78]	COMMERCIAL
CPLEX [79]	COMMERCIAL
XPRESS [80]	COMMERCIAL
MOSEK [81]	COMMERCIAL
LGO [82]	COMMERCIAL
KNITRO [83]	COMMERCIAL
SCIP [84]	FREE / OPEN-SOURCE
Google Optimization tools [85]	FREE / OPEN-SOURCE
COIN-OR [86] (IPOPT, CLP, ...)	FREE / OPEN-SOURCE
GLPK [87]	FREE / OPEN-SOURCE
SNOPT [88]	FREE / OPEN-SOURCE
MINOS [89]	FREE / OPEN-SOURCE
LP_SOLVE [90]	FREE / OPEN-SOURCE
MIPCL [91]	FREE / OPEN-SOURCE

receiving the solution vector. Modelers are unavoidable in non-trivial applications, to avoid the error-prone process of manually building *solver* inputs and processing their outputs, respecting each proprietary solver conventions. Each modeler typically comes with its own human-readable syntax for problem description, although can also accept descriptions in widely-used modeling languages like mathematical programming systems (MPS), named after an early IBM linear programming product which became the de-facto standard [62].

Table III lists a number of available modelers, both open-source and commercial. External modelers come with its own front-end, so the main program with the business logic should run on it, or communicate with it via file passing. In its turn, library-based modelers integrate in the main program where the problem data is, and thus permit a simpler workflow. Note that passing via files the problem inputs can be significantly inefficient in large-scale problems. Therefore, math-heuristic approaches, where successive calls to the solver are performed in algorithm iterations, should be implemented with fast library-based modelers. A math-heuristic algorithm for large-scale fault tolerant WDM plant design, implemented using the JOM library in Net2Plan, is reported in [60], where approx. ten thousand invocations to the solver are completed in a global running time of one hour.

2) Solvers: Table IV lists the main *solvers* available, both commercial and open-source. Not all solvers are applicable for all types of problems, the references provide the most updated capabilities in each suite. Regarding MILPs, the most widely used solvers in network design, it is useful to follow the benchmark solvers' tests like [63], to comparatively evaluate their merits.

Although not shown in Table III, most commercial solvers listed include internal and/or external modelers, to interface with them from Java, C++ and/or Python programs. Unfortunately, free solvers seldom do that, relying instead on external modelers, or ad-hoc provided libraries.

IV. NETWORK OPTIMIZATION TOOLS

A network optimization tool is a software combining two base capabilities: (i) a network simulation engine, able to predict IP/MPLS/BGP and (ideally) multilayer network reactions under e.g., failures or congestions, and (ii) resource allocation and dimensioning algorithms to plan and optimize the network under user-defined requisites in terms of e.g., QoS and fault tolerance, that rely on the network simulation engine to optimize the network while SLAs are met.

Network optimization tools differ from general purpose network simulators like NS-3 [92], and OMNet++ [93], which provide an event-driven simulation engine with libraries that reproduce network protocols (IP, Ethernet, MPLS . . .) *at the packet level*. However, these network simulators do not code flow-level network reactions and have no support for the network optimization side e.g., capacity planning, service provisioning.

Several vendors have acquired former third party network optimization and planning tools, integrating them into their portfolios. This is the case of Cisco WAE Design [94], Juniper IP/MPLS View [95], and Packet Design [96], acquired by Ciena in 2018. Some of the commercial third party alternatives are OPNET [97] (now part of Riverbed portfolio) and E-lighthouse Network Planner [98]. The latter is a product of a startup of Net2Plan founders, a commercial alternative providing a full IP/MPLS/BGP over DWDM *multilayer* planning tool, covering from a full OTN optimization with optical impairments calculations, to IP traffic engineering and traffic forecasting.

Network optimization tools have roadmaps to extend its reach to cover (i) an operation based on SDN controllers to interact with the hardware, and (ii) NVF optimization, e.g., Cisco's Optimization Service for NFV that offers, among the inherent vendor-specific programs, OpenStack (i.e., open-source) Lifecycle Management support [99].

A. *Open Source Network Optimization. Net2Plan*

Network optimization and planning is a well-established research field in optical and IP technologies, and a good amount of open-source versions of the research algorithmic proposals can be found in repositories like GitHub. However, these are contributions fitted to a specific network problem in a specific technology, not a *framework*. That is, they lack aspects like a Graphical User Interface (GUI), a reasonable software maturity, applicability to multiple technologies or a support for integrating algorithms for different use cases.

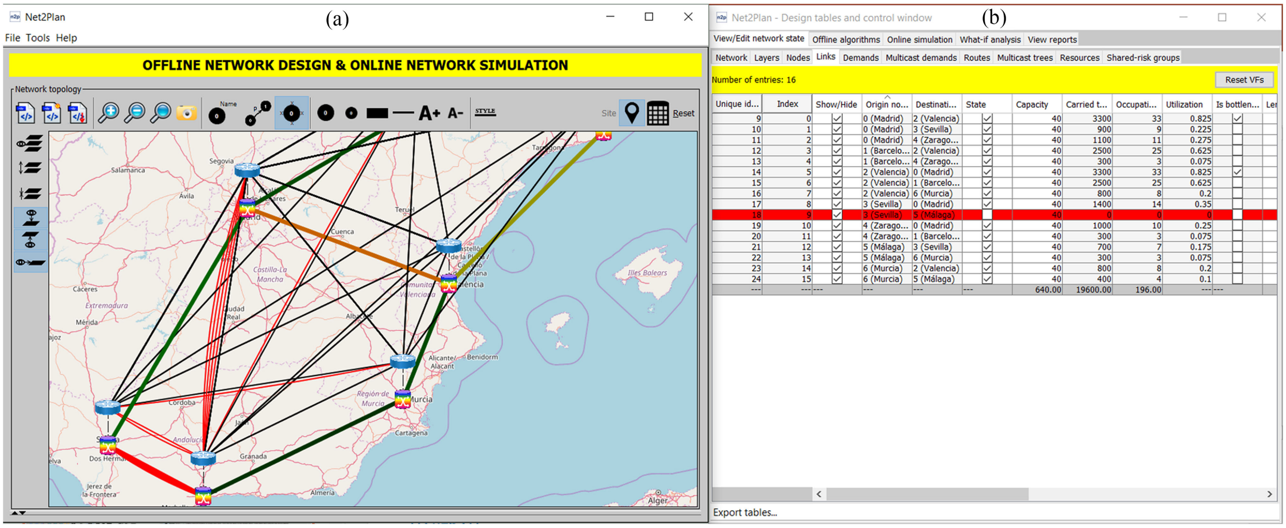


Fig. 6. (a) Network reaction at different layer to a manually created failure/repair and/or traffic shifts. (b) Selection of the link under evaluation. (Image from [1]).

To the best of the authors' knowledge, the only network optimization/planning *framework* matching these minimum requirements is Net2Plan [11]. Net2Plan is a free and open-source Java tool initiated in 2011, with its last major release (0.6.0) delivered in July 2018. Net2Plan is devoted to the planning, optimization and evaluation of communication networks [11], with the following major characteristics:

- *Multilayer technology-agnostic network representation*, that permits adapting Net2Plan to multiple network technologies (e.g., examples for WDM networks, IP networks, IP over WDM networks, wireless ad-hoc and wireless cellular networks has been described in [60] and are available in [100]).
- *Open API* for developing and plugging-in self-made or already-available optimization algorithms.
- *Open source* and accessible in usual repositories (e.g., github) [100].
- *Command line interface (CLI) and GUI* for network visualization and manipulation of the data.
- *Scalability*. Net2Plan has been tested with >10 K nodes.

Net2Plan has been designed as an intrinsic multi-layer network planning/optimization tool [101]. Consequently, its abstract model contains the concepts of *layer*, *node*, *link*, *demand*, *multicast demand, multicast tree*, *route*, *shared-risk-group*, etc. Technology related information can be attached to the elements via *attributes*, and *tags*, that Net2Plan kernel does not process, but plugin algorithms can utilize. For instance, an algorithm can use an attribute called *wavelength* in the *route* objects to store the spectrum occupation of the lightpaths.

Net2Plan GUI provides a multilayer visualization (e.g., see Fig. 6) geopositioning the nodes. Per link, node, demand etc. information is summarized in tables that show performance statistics. The user can introduce failure and congestion reactions of the network via built-in or user-developed plugins e.g., for coding IP over WDM behaviors and how failures propagate among layers for different network recovery schemes. Then, the GUI *what-if* analysis functionality permits e.g., manually failing links and observing the network reaction in the tables (as in Fig. 6).

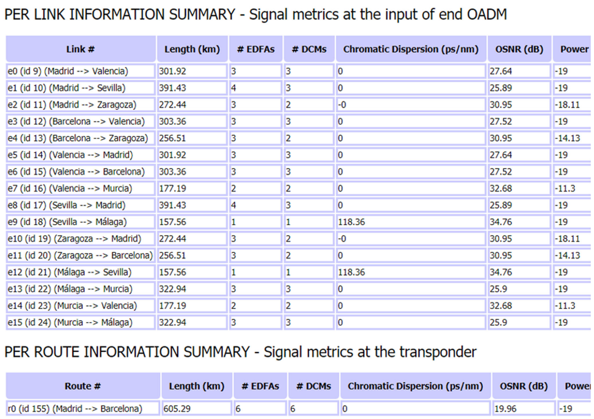
Net2Plan permits plugging-in (i) offline algorithms, (ii) online algorithms, and (iii) reports:

- *Offline* algorithms are mainly devoted to capacity planning and network designs: the algorithm receives the current network design, as well as the user-defined input parameters, and returns a new design.
- *Online* algorithms speed-up the process of building event-driven simulations for statistically testing the systems. The user can plug-in (a) an event-generation module coding how failures, repairs and traffic fluctuations occur in the network along time, and (b) the module used in *what-if* analysis for coding network reactions to previous events. Net2Plan permits running and tracing the simulation in the GUI and obtaining statistical results.
- *Reports* are user-defined or built-in plugins that produce an HTML output, from the current design.

Net2Plan includes an extensive built-in repository of offline planning and online provisioning algorithms, as well as reports, documented in a Javadoc format [102], with links to sections in [60] describing the mathematical details. In particular, built-in algorithms address as constrained optimization formulations the essential network design problems appearing in any network technology: routing the traffic; capacity assignment to the links; controlling the source rates (i.e., congestion control); and deciding the network topology. Furthermore, the mathematical techniques for these computer network problems are also available and described, ranging from gradient-based algorithms suitable, for example, for capacity planning, up to a dual decomposition approach for cross-layer congestion control. Additionally, some of these resources are of direct application to the SDN/NFV use cases described in Section II, and are reviewed in the next Subsection IV.E.

Net2Plan includes the following functionalities for modeling traffic demands in the network:

- *Synthesis and normalization of traffic matrices:* The user can use realistic models for building traffic matrices, based



PER LINK INFORMATION SUMMARY - Signal metrics at the input of end OADM

Link #	Length (km)	# EDFAs	# DCMs	Chromatic Dispersion (ps/nm)	OSNR (dB)	Power
e0 (id 9) (Madrid --> Valencia)	301.92	3	3	0	27.64	-19
e1 (id 10) (Madrid --> Sevilla)	391.43	4	3	0	25.89	-19
e2 (id 11) (Madrid --> Zaragoza)	272.44	3	2	-0	30.95	-18.11
e3 (id 12) (Barcelona --> Valencia)	303.36	3	3	0	27.52	-19
e4 (id 13) (Barcelona --> Zaragoza)	256.51	3	2	0	30.95	-14.13
e5 (id 14) (Valencia --> Madrid)	301.92	3	3	0	27.64	-19
e6 (id 15) (Valencia --> Barcelona)	303.36	3	3	0	27.52	-19
e7 (id 16) (Valencia --> Murcia)	177.19	2	2	0	32.68	-11.3
e8 (id 17) (Sevilla --> Madrid)	391.43	4	3	0	25.89	-19
e9 (id 18) (Sevilla --> Málaga)	157.56	1	1	118.36	34.76	-19
e10 (id 19) (Zaragoza --> Madrid)	272.44	3	2	-0	30.95	-18.11
e11 (id 20) (Zaragoza --> Barcelona)	256.51	3	2	0	30.95	-14.13
e12 (id 21) (Málaga --> Sevilla)	157.56	1	1	118.36	34.76	-19
e13 (id 22) (Málaga --> Murcia)	322.94	3	3	0	25.9	-19
e14 (id 23) (Murcia --> Valencia)	177.19	2	2	0	32.68	-11.3
e15 (id 24) (Murcia --> Málaga)	322.94	3	3	0	25.9	-19

PER ROUTE INFORMATION SUMMARY - Signal metrics at the transponder

Route #	Length (km)	# EDFAs	# DCMs	Chromatic Dispersion (ps/nm)	OSNR (dB)	Powe
r0 (id 155) (Madrid --> Barcelona)	605.29	6	6	0	19.96	-19

Fig. 7. Automatic report containing WDM line engineering information per link and per lightpath: OSNR (using GN model), PMD, CD, optical power [1]. (Image from [1]).

e.g., node population-distance relations, and then manipulate and normalize those matrices in different forms.

- *Monitoring information:* The user can import and export monitoring information of links and/or demands, consisting of time series of traffic. Simple prediction models can then be applied to them.
- IP *traffic matrix estimation:* Simple models (e.g., gravity model) are included to derive the traffic injected by the IP unicast and multicast traffic demands, from the link monitoring information.

Net2Plan flexibility makes it applicable to a variety of scenarios. In the following sections, we describe the open-source efforts to apply Net2Plan in some of the SDN/NFV use cases described in Section II.

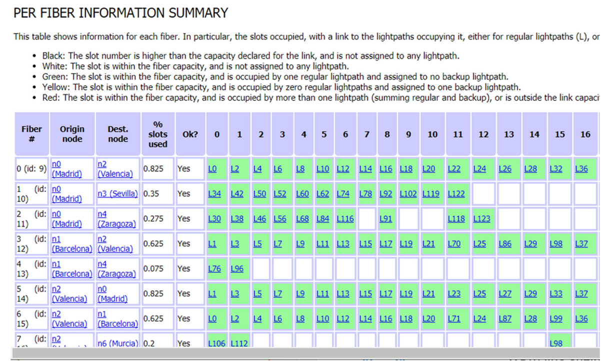
B. Vendor Agnostic WDM Line Engineering

Net2Plan addresses the use case "optical-layer transport control" described in Section II-A incorporating a *WDM line engineering* report functionality, a useful resource for the planning, optimization and evaluation of multi-vendor WDM line systems [103].

The report is produced from a built-in IP over WDM network representation where the user can define WDM links with some fiber, line amplifiers and OADM characteristics, and optical connections with their paths and transponder/regenerator characteristics.

The report (Fig. 7) produces per WDM link and per lightpath information, estimating power budgets, OSNR, chromatic and polarization mode dispersions (CD and PMD) from the input information. Warnings are raised when e.g., optical power is outside amplifier or transponder ranges, or OSNR/CD/PMD values are outside transponder capabilities.

For OSNR computations, the contribution of nonlinear impairments was originally computed as manually set OSNR penalties. More recently, the wide acceptance and experimental validation of the GN model [36], and the collaboration within TIP-OOPT working group, has motivated its inclusion in Net2Plan [104]. In particular, the Net2Plan extension available in [104] interacts with the GNPY library [35] to estimate



PER FIBER INFORMATION SUMMARY

This table shows information for each fiber. In particular, the slots occupied, with a link to the lightpaths occupying it, either for regular lightpaths (L), or

- Black: The slot number is higher than the capacity declared for the link, and is not assigned to any lightpath.
- White: The slot is within the fiber capacity, and is not assigned to any lightpath.
- Green: The slot is within the fiber capacity, and is occupied by one regular lightpath and assigned to no backup lightpath.
- Yellow: The slot is within the fiber capacity, and is occupied by zero regular lightpaths and assigned to one backup lightpath.
- Red: The slot is within the fiber capacity, and is occupied by more than one lightpath (summing regular and backup), or is outside the link capaci

Fiber #	Origin node	Dest. node	% slots used	Ok?	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0 (id: 9)	n0 (Madrid)	n2 (Valencia)	0.825	Yes	L0	L2	L4	L6	L8	L10	L12	L14	L16	L18	L20	L22	L24	L26	L28	L32	L36
1 (id: 10)	n0 (Madrid)	n3 (Sevilla)	0.35	Yes	L34	L42	L50	L52	L60	L62	L74	L78	L92	L102	L119	L122					
2 (id: 11)	n0 (Madrid)	n4 (Zaragoza)	0.275	Yes	L30	L38	L46	L56	L68	L84	L116		L91			L118	L123				
3 (id: 12)	n1 (Barcelona)	n2 (Valencia)	0.625	Yes	L1	L3	L5	L7	L9	L11	L13	L15	L17	L19	L21	L70	L25	L86	L29	L98	L37
4 (id: 13)	n1 (Barcelona)	n4 (Zaragoza)	0.075	Yes	L76	L96															
5 (id: 14)	n2 (Valencia)	n0 (Madrid)	0.825	Yes	L1	L3	L5	L7	L9	L11	L13	L15	L17	L19	L21	L23	L25	L27	L29	L33	L37
6 (id: 15)	n2 (Valencia)	n1 (Barcelona)	0.625	Yes	L0	L2	L4	L6	L8	L10	L12	L14	L16	L18	L20	L71	L24	L87	L28	L99	L36
7 (id:	n2	n6 (Murcia)	0.2	Yes	L106	L112														L98	

Fig. 8. Routing and Spectrum Assignment (RSA) inspector showing, for each fiber, the slots occupied with a link on each slot to the lightpath occupying it. (Image from [1]).

the nonlinear contribution to the optical impairments in WDM networks using the GN model.

C. IP Over WDM Optimization via SDN Controllers

To address the use case discussed in Section II-B, Net2Plan permits interfacing with SDN controllers as recently reported in three proofs of concept. First, [105] reports Net2Plan interaction with the OpenDaylight SDN controller. Second, [106] demonstrates the usage of a stateful PCE as external module of a Net2Plan multi-layer dynamic IP/MPLS-over-WDM scenario. Third, Net2Plan and ONOS interaction is reported in two works: a preliminary approach based on ad-hoc extensions and an intermediate representation model based on NetRap [107], and later in Section IV.D a demonstration of an ONOS client based on it API allowing the interaction with Net2Plan [57].

The previous works prototype the application of Net2Plan sitting at the NBI of the controller, receiving topology and traffic information, and providing OaaS functionalities with it.

Once connected to the SDN controller, the full optimization capabilities of Net2Plan are available. The following built-in functionalities are highlighted in the context of IP over WDM networks:

- *Fault-tolerant IP over WDM design algorithms:* Net2Plan provides built-in algorithms for determining the virtual topology, routing and spectrum assignment (RSA) of the lightpaths and routing of the IP flows, in flexi-grid IP over WDM networks, for different network recovery schemes (e.g., using not protected or 1 + 1 protected lightpaths), and different mixed transponder modulations and optical reaches. The fault tolerance target is user-defined using the SRG (Shared-Risk-Group) concept, that represents a network vulnerability (e.g., a duct cut) and the associated set of resources that simultaneously fail when such risk materializes (all the traversing fibers are cut).
- *RSA inspector report:* This is a Net2Plan report that automatically generates a table-format visualization of the spectrum resources occupied in a design, as shown in Fig. 8. In particular, the RSA inspector lists the slots occupied for each fiber and if they are assigned to regular lightpaths (L) or protection segments (P). Additionally, a color code permits an additional classification considering the fiber capacity, i.e., depending on the number of slots

Layer WDM, index = 0, id = 1

Unicast traffic

SRG Index failed	Offered traffic	Blocked traffic (%)	Offered traffic traversing oversubscribed links (%)	Offered traffic of demands with excessive latency (%)	Total blocked traffic [out of contract] (%)	% of demands fully ok
No failure	13000.000	0.000 (0.000 %)	0.000 (0.000 %)	0.000 (0.000 %)	0.000 (0.000 %)	(100.000 %)
0	13000.000	6600.000 (50.769 %)	0.000 (0.000 %)	0.000 (0.000 %)	6600.000 (50.769 %)	(49.231 %)
1	13000.000	2800.000 (21.538 %)	0.000 (0.000 %)	0.000 (0.000 %)	2800.000 (21.538 %)	(78.462 %)
2	13000.000	2200.000 (16.923 %)	0.000 (0.000 %)	0.000 (0.000 %)	2200.000 (16.923 %)	(83.077 %)
3	13000.000	5000.000 (38.462 %)	0.000 (0.000 %)	0.000 (0.000 %)	5000.000 (38.462 %)	(61.538 %)
4	13000.000	600.000 (4.615 %)	0.000 (0.000 %)	0.000 (0.000 %)	600.000 (4.615 %)	(95.385 %)
5	13000.000	1600.000 (12.308 %)	0.000 (0.000 %)	0.000 (0.000 %)	1600.000 (12.308 %)	(87.692 %)
6	13000.000	1400.000 (10.769 %)	0.000 (0.000 %)	0.000 (0.000 %)	1400.000 (10.769 %)	(89.231 %)
7	13000.000	800.000 (6.154 %)	0.000 (0.000 %)	0.000 (0.000 %)	800.000 (6.154 %)	(93.846 %)

Layer IP, index = 1, id = 285

Unicast traffic

SRG Index failed	Offered traffic	Blocked traffic (%)	Offered traffic traversing oversubscribed links (%)	Offered traffic of demands with excessive latency (%)	Total blocked traffic [out of contract] (%)	% of demands fully ok
No failure	10000.000	0.000 (0.000 %)	0.000 (0.000 %)	0.000 (0.000 %)	0.000 (0.000 %)	(100.000 %)
0	10000.000	0.000 (0.000 %)	9320.197 (93.202 %)	0.000 (0.000 %)	9320.197 (93.202 %)	(33.333 %)
1	10000.000	0.000 (0.000 %)	7727.512 (77.275 %)	0.000 (0.000 %)	7727.512 (77.275 %)	(40.476 %)
2	10000.000	0.000 (0.000 %)	5605.399 (56.054 %)	0.000 (0.000 %)	5605.399 (56.054 %)	(61.905 %)
3	10000.000	0.000 (0.000 %)	6503.431 (65.034 %)	0.000 (0.000 %)	6503.431 (65.034 %)	(54.762 %)

Fig. 9. Availability analysis in which columns show: SRG index, offered traffic, blocked traffic (%), offered traffic traversing oversubscribed links (%), offered traffic of demands with excessive latency (%), total blocked traffic [out of contract] (%), % of demands fully ok. (Image from [1]).

occupied, their regular/backup belongingness and the fiber capacity.

- *Availability analysis:* Net2Plan provides two types of reports for analyzing the network vulnerabilities. First, a report that automatically simulates the network reaction for the user-defined risks (SRG), summarizing the network survivability for each of them at the WDM and IP level. Fig. 9 illustrates this report, showing for each risk the offered and blocked traffic, as well as the traffic traversing oversubscribed links or with excessive end-to-end latency (computed considering true multilayer traffic propagation). Second, NetPlan includes a built-in report that makes use of the Mean-Time-To-Fail (MTTF), and Mean-Time-To-Repair (MTTR) information associated to each vulnerability (SRG), to provide a statistical availability of the network, including an error margin.

D. NFV Optimization

Two Net2Plan prototypes have been developed for interfacing and optimizing OpenStack systems using the OpenStack4j [108] client for Java. First, in [17], [57] a Net2Plan GUI was developed to interface with OpenStack, as a part of the joint IT and network optimization efforts described in next section. Second, [52] is an ongoing project consisting of a Net2Plan GUI for visualizing and controlling an OpenStack, in those aspects related to identity management (*Keystone* service), computing resources (*Nova* service), network resources (*Neutron* service) and performance monitoring (*Ceilometer* service).

The target of these efforts is enabling Net2Plan as a framework for plugging-in user-developed algorithms, to optimize OpenStack systems, which address the use cases in Section II-C.

E. Joint IT and Multi-Layer Network Resource Provisioning

Net2Plan prototype in [57] demonstrates a connection from Net2Plan to (i) an OpenSource MANO system, (ii) a number of OpenStack data centers, and (iii) an SDN ONOS controller managing the transport network interconnecting the data centers.

Fig. 10 shows the workflow demonstrated in [57]. The user specifies a network service to instantiate in Net2Plan, indicating the IP flow initial and destination node, its rate (Gbps), its end-to-end latency requisites, and the sequence of the types of VNFs to be traversed (e.g., first a vFirewall, then a Network Address Translator, then a video transcoder). The amount of CPU/RAM/HD consumed and average processing time added, is also user-defined for each VNF type. Net2Plan is aware of the IT resources available via its direct interface to the OpenStacks, and is aware of the network resources via its interface with ONOS. Then, for each network service request, it jointly optimizes (i) the VNF placement (in which data center to instantiate each VNF), and (ii) the path in the transport network connecting the flow end nodes and the VNFs' data centers. Therefore, given that Net2plan receives the entire picture of the system, both IT and network resources are configured by the NVF orchestrator and the SDN controller after the optimization, being the latter in charge of setting up the devices/agents of the optical level.

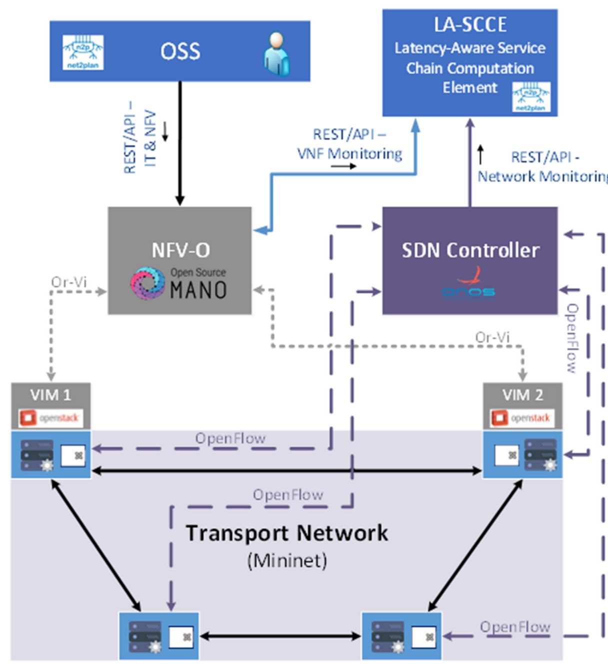


Fig. 10. Functional architecture of the joint transport and IT resource allocation optimization demonstration [57].

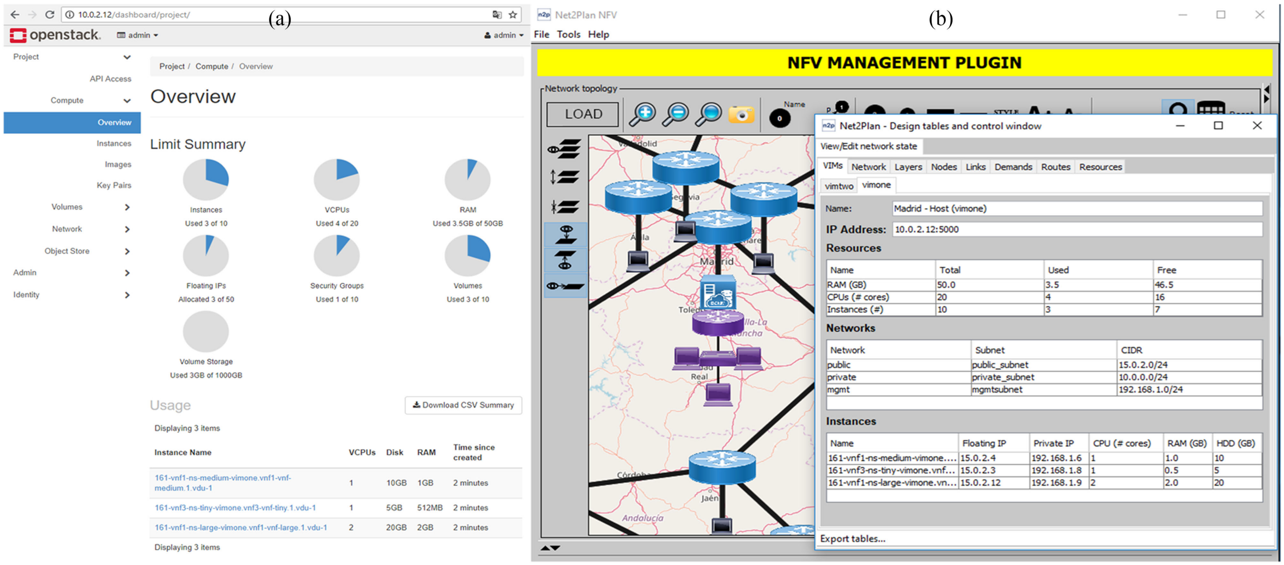


Fig. 11. (a) OpenStack dashboard showing three running VNF instances: small, medium and large, respectively. (b) Net2Plan GUI implementing the NFV management plugin that communicates with OpenStack, showing the three VNF instances on an OpenStack domain (network) emulated as a node in Madrid. (Image from [1]).

Fig. 11 illustrates a screenshot of the demonstrated functionalities. Fig. 11(a) shows the OpenStack (GUI) dashboard reporting the occupied resources in the data center, by the VNF instances. Fig. 11(b) displays information present in Net2Plan GUI: a representation of both the physical network read from ONOS (blue) and the virtual network of VMs instantiated inside the OpenStacks (purple), with per VM information like name, floating and private IP addresses, CPU, RAM, HD, etc.

Net2Plan provides resources for fast prototyping of user-defined algorithms to jointly allocate IT and network resources in this SDN/NFV context. In particular:

- *NFV modeling:* Net2Plan includes since version 0.5.0 released on 15 Mar 2017, the possibility to model nodes with IT resources (e.g., CPU, RAM, HD), that can host VNF instances consuming those resources, and then be part of service chains. Net2Plan API includes libraries to ease the manipulation of these objects.
- *Offline algorithms:* Net2Plan includes built-in algorithms that solve the service chain placement problem with both a heuristic and an ILP [11]. The algorithm receives (i) the service chain requests, with the end nodes, requested capacity and sequence of VNF types to traverse, (ii) the IT resources and network links available, and (iii) the requisites in terms of CPU/RAM/HD of each VNF instance, as well as its processing capacity. As an output, the algorithm decides (i) what VNF instances to create and where, and (ii) the sequence of links and VNF instances followed by each service chain. Another heuristic algorithm has been presented in [57], with the enhanced functionality of providing allocations subject to end-to-end latency considerations. Note that multiple schemes for IP over WDM resource provisioning can be jointly used with the IT resources management and allocation. For instance, a number of candidate shortest path can be searched in the network and then evaluate them according to a given requirement (e.g., bandwidth or latency). Indeed, a load-balancing RSA strategy could be implemented jointly with the consideration of the IT resources allocation and availability.
- *kMCSC library:* Net2Plan offers a built-in utility library implementing the k minimum cost service chain problem (kMCSC), that can be useful for fast prototyping capacity planning or provisioning algorithms. The kMCSC algorithm receives the input and output nodes of a chain, the sequence of VNF types to traverse, and the cost associated to each link or VNF instance, and returns a ranking of the k minimum cost service chain realizations. The algorithms mentioned in the previous item make use of the kMCSC routine.

V. Summary

The combination of SDN and NFV is fostering transport ecosystems characterized by unprecedented network control and unique resource dynamicity. These ecosystems cannot afford traditional manual optimization approaches. Instead, leveraging on the emergence of numerous open systems that control, manage and orchestrate SDN/NFV-enabled networks, programming abstractions can be exploited by third-party applications sitting at the NBI of SDN controllers and SDN/NFV orchestrators, following OaaS business models.

In this paper, we overviewed open-source optimization software initiatives for offline planning, online provisioning and orchestration of SDN/NFV networks. We first discussed a set of realistic use cases where new optimization challenges appear. Then, we discussed different mathematical approaches for developing the algorithms at the heart of the optimization engines. We concentrated on the open-source *modelers* and *solvers* available, and discuss different aspects of its application. Finally we focused on Net2Plan, a complete open-source network optimization framework. We reviewed recent works that illustrate its potentials as a network optimization NBI application, on top of SDN controllers and/or NFV orchestrators.

References

[1] P. P. Mariño, M. Garrich, and F. J. Moreno-Muro, "The role of open-source network optimization software in the SDN/NFV world," in *Proc. Opt. Fiber Commun. Conf. Expo.*, San Diego, CA, USA, 2018, pp. 1–71.

[2] Cisco visual networking index: Forecast and methodology, pp. 2016–2021. White Paper, Sep. 2017. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html

[3] 5G infrastructure public private partnership (5G PPP) key performance indicators (KPIs). [Online]. Available: https://5g-ppp.eu/kpis/. Accessed on: Sep. 15, 2018.

[4] Open Networking Foundation, "Software-defined networking: The new norm for networks," White Paper, Apr. 2012. [Online]. Available: https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf

[5] D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015.

[6] M. Chiosi *et al.*, "Network functions virtualisation: An introduction, benefits, enablers, challenges and call for action," in *Proc. SDN OpenFlow World Congr.*, Darmstadt, Germany, Oct. 2012, pp. 22–24. [Online]. Available: http://portal.etsi.org/NFV/NFV_White_Paper.pdf

[7] J. G. Herrera and J. F. Botero, "Resource allocation in NFV: A comprehensive survey," *IEEE Trans. Netw. Service Manage.*, vol. 13, no. 3, pp. 518–532, Sep. 2016.

[8] OpenStack. [Online]. Available: https://www.openstack.org/. Accessed on: Sep. 15, 2018.

[9] Open network operating system (ONOS). [Online]. Available: https://onosproject.org/. Accessed on: Sep. 15, 2018.

[10] Open source mano. [Online]. Available: https://osm.etsi.org/. Accessed on: Sep. 15, 2018.

[11] Net2Plan. [Online]. Available: http://www.net2plan.com/. Accessed on: Sep. 15, 2018.

[12] OpenROADM. [Online]. Available: http://www.openroadm.org/. Accessed on: Sep. 15, 2018.

[13] OpenConfig. [Online]. Available: http://www.openconfig.net/. Accessed on: Sep. 15, 2018.

[14] Open and disaggregated transport network (ODTN). [Online]. Available: https://www.opennetworking.org/solutions/odtn/. Accessed on: Sep. 15, 2018.

[15] C. Qiaogang *et al.*, "Functional requirements for transport API, open networking foundation, technical recommendations ONF TR-527," Open Netw. Found., Palo Alto, CA, USA, Tech. Rep. ONF TR-527, 2016. [Online]. Available: https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/TR-527_TAPI_Functional_Requirements.pdf

[16] Open network automation platform (ONAP). [Online]. Available: https://www.onap.org/. Accessed on: Sep. 15, 2018.

[17] F. J. Moreno-Muro *et al.*, "Joint optimal service chain allocation, VNF instantiation and metro network resource management demonstration," in *Proc. Opt. Fiber Commun. Conf. Expo.*, San Diego, CA, USA, 2018, pp. 1–3.

[18] E. Riccardi, P. Gunning, O. G. de Dios, M. Quagliotti, V. López, and A. Lord, "An operator's view on introduction of white boxes in optical networks," *J. Lightw. Technol.*, vol. 36, no. 15, pp. 3062–3072, Aug. 2018.

[19] R. Enns *et al.*, "Network configuration protocol (NETCONF)," Internet Engineering Task Force, Fremont, CA, USA, IETF RFC 6241, Jun. 2011.

[20] M. Bjorklund, "YANG—A data modeling language for the network configuration protocol (NETCONF)," Internet Engineering Task Force, Fremont, CA, USA, IETF RFC 6020, 2010.

[21] M. Garrich and A. Bravalheri, "Overview of South-Bound interfaces for software-defined optical networks," in *Proc. Int. Conf. Transparent Opt. Netw.*, Bucharest, Romania, 2018, pp. 1–4.

[22] M. Dallaglio, N. Sambo, J. Akhtar, F. Cugini, and P. Castoldi, "YANG model and NETCONF protocol for control and management of elastic optical networks," in *Proc. Opt. Fiber Commun. Conf. Exhib.*, Anaheim, CA, USA, 2016, pp. 1–3.

[23] J. Akhtar, "YANG modeling of network elements for the management and monitoring of elastic optical networks," in *Proc. IEEE Int. Conf. Telecommun. Photon.*, Dhaka, Bangladesh, 2015, pp. 1–5.

[24] M. Dallaglio, N. Sambo, F. Cugini, and P. Castoldi, "Management of sliceable transponder with NETCONF and YANG," in *Proc. Int. Conf. Opt. Netw. Des. Model.*, Cartagena, Spain, 2016, pp. 1–6.

[25] M. Dallaglio, N. Sambo, F. Cugini, and P. Castoldi, "Pre-programming resilience schemes upon failure through NETCONF and YANG," in *Proc. Opt. Fiber Commun. Conf. Exhib.*, Los Angeles, CA, USA, 2017, pp. 1–3.

[26] A. Sgambelluri *et al.*, "Fully disaggregated ROADM white box with NETCONF/YANG control, telemetry, and machine learning-based monitoring," in *Proc. Opt. Fiber Commun. Conf. Expo.*, San Diego, CA, USA, 2018, pp. 1–3.

[27] O. F. Yilmaz *et al.*, "Automated management and control of a multivendor disaggregated network at the L0 Layer," in *Proc. Opt. Fiber Commun. Conf. Expo.*, San Diego, CA, USA, 2018, pp. 1–3.

[28] Lumentum Inc., "Transport ROADM Whitebox/Graybox," [Online]. Available: https://www.lumentum.com/en/optical-communications/products/sdn-whiteboxes-and-grayboxes. Accessed on: Sep. 15, 2018.

[29] Fujitsu, "Network evolution with 1FINITY series," [Online]. Available: http://www.fujitsu.com/global/products/network/products/1finity/. Accessed on: Sep. 15, 2018.

[30] Juniper Networks Inc., "TCX1000 programmable ROADM," [Online]. Available: https://www.juniper.net/us/en/products-services/packet-optical/tcx-series/tcx1000/. Accessed on: Sep. 15, 2018.

[31] Telecom infrastructure project (TIP). [Online]. Available: http://telecominfraproject.com/. Accessed on: Sep. 15, 2018.

[32] Open optical & packet transport (OOPT) group. [Online]. Available: http://telecominfraproject.com/open-optical-packet-transport/. Accessed on: Sep. 15, 2018.

[33] I. Lyubomirsky, B. Taylor, and H.-J. W. Schmidtke, "An open approach for switching, routing, and transport," Nov. 2016. [Online]. Available: https://code.facebook.com/posts/1977308282496021/an-open-approach-for-switching-routing-and-transport/

[34] H.-J. Schmidtke and L. M. Garcia, "Driving openness in optical networks: An update from the OOPT project," Nov. 2017. [Online]. Available: http://telecominfraproject.com/riving-openness-in-optical-networks-an-update-from-the-oopt-project-group/

[35] GNPY library documentation. [Online]. Available: https://gnpy.readthedocs.io. Accessed on: Sep. 15, 2018.

[36] P. Poggiolini, "The GN model of nonlinear propagation in uncompensated coherent optical systems," *J. Lightw. Technol.*, vol. 30, no. 24, pp. 3857–3879, Dec. 2012.

[37] G. Grammel, V. Curri, and J. Auge, "Physical simulation environment of the telecommunications infrastructure project (TIP)," in *Proc. Opt. Fiber Commun. Conf. Expo.*, San Diego, CA, USA, 2018, pp. 1–3.

[38] M. Filer *et al.*, "Multivendor experimental validation of an open source QoT estimator for optical networks," *J. Lightw. Technol.*, vol. 36, no. 15, pp. 3073–3082, Aug. 2018.

[39] M. De Leenheer, Y. Higuchi, and G. Parulkar, "An open controller for the disaggregated optical network," in *Proc. Int. Conf. Opt. Netw. Des. Model.*, Dublin, Ireland, May 2018, pp. 230–233.

[40] T. Szyrkowiec, A. Autenrieth, and W. Kellerer, "Optical network models and their application to software-defined network management," *Int. J. Opt.*, vol. 2017, 2017, Art. no. 5150219.

[41] O. Gerstel, "Control architectures for multilayer networking: Distributed, centralized, or something in between?," in *Proc. Opt. Fiber Commun. Conf. Exhib.*, Los Angeles, CA, USA, 2015, pp. 1–16.

[42] OpenStack's compute service project Nova. [Online]. Available: https://www.openstack.org/software/releases/queens/components/nova. Accessed on: Sep. 15, 2018.

[43] The filter scheduler in Nova. [Online]. Available: https://docs.openstack.org/nova/latest/user/filter-scheduler.html. Accessed on: Sep. 15, 2018.

[44] Availability zones in OpenStack. [Online]. Available: https://docs.openstack.org/newton/networking-guide/config-az.html. Accessed on: Sep. 15, 2018.

[45] CPU topologies definition in OpenStack. [Online]. Available: https://docs.openstack.org/nova/pike/admin/cpu-topologies.html. Accessed on: Sep. 15, 2018.

[46] M. Scharf, M. Stein, T. Voith, and V. Hilt, "Network-aware instance scheduling in OpenStack," in *Proc. IEEE Int. Conf. Comput. Commun. Netw.*, Las Vegas, NV, USA, 2015, pp. 1–6.

[47] OpenStack's orchestration project Heat. [Online]. Available: https://www.openstack.org/software/releases/queens/components/heat. Accessed on: Sep. 15, 2018.

[48] Autoscaling groups in OpenStack Heat. [Online]. Available: https://docs.openstack.org/heat/pike/template_guide/openstack.html#OS::Heat::AutoScalingGroup. Accessed on: Sep. 15, 2018.

[49] Amazon web services (AWS). [Online]. Available: https://aws.amazon.com/es/. Accessed on: Sep. 15, 2018.

[50] OpenStack's optimization service project Watcher. [Online]. Available: https://www.openstack.org/software/releases/queens/components/watcher. Accessed on: Sep. 15, 2018.

[51] Watcher architecture. [Online]. Available: https://docs.openstack.org/watcher/pike/architecture.html. Accessed on: Sep. 15, 2018.

[52] Net2Plan-OpenStack project. [Online]. Available: https://github.com/girtel/Net2Plan-OpenStack. Accessed on: Sep. 15, 2018.

[53] Central Office Re-architected as a Datacenter (CORD) project. [Online]. Available: https://opencord.org/. Accessed on: Sep. 15, 2018.

[54] P. Pavon-Marino, F. J. Moreno-Muro, and N. Skorin-Kapov, "Evolution of core traffic for growing CDNs: Is the growth rate of core network traffic overestimated?," in *Proc. Opt. Fiber Commun. Conf. Exhib.*, Los Angeles, CA, USA, 2017, pp. 1–16.

[55] ETSI open source NFV management and orchestration (MANO). [Online]. Available: https://www.etsi.org/technologies-clusters/technologies/nfv/open-source-mano. Accessed on: Sep. 15, 2018.

[56] R. Casellas *et al.*, "Metro-haul: SDN control and orchestration of disaggregated optical networks with model-driven development," in *Proc. Int. Conf. Transparent Opt. Netw.*, Bucharest, Romania, 2018, pp. 1–4.

[57] F. J. Moreno-Muro *et al.*, "Latency-aware optimization of service chain allocation with joint VNF instantiation and SDN metro network control," in *Proc. Eur. Conf. Opt. Commun.*, Rome, Italy, 2018, pp. 1–3.

[58] OpenStack's NFV Orchestration project Tacker. [Online]. Available: https://www.openstack.org/software/releases/queens/components/tacker. Accessed on: Sep. 15, 2018.

[59] The Linux foundation networking. [Online]. Available: https://www.linuxfoundation.org/projects/networking/. Accessed on: Sep. 15, 2018.

[60] P. Pavón-Mariño, *Optimization of Computer Networks: Modeling and Algorithms: A Hands-on Approach*. Hoboken, NJ, USA: Wiley, 2016.

[61] B. Garcia-Manrubia *et al.*, "Offline impairment-aware RWA and regenerator placement in translucent optical networks," *J. Lightw. Technol.*, vol. 29, no. 3, pp. 265–277, Feb. 2011.

[62] Optimization subroutine library (IBM). [Online]. Available: http://publibfp.dhe.ibm.com/cgi-bin/bookmgr/BOOKS/ekkosl01/CCONTENTS. Accessed on: Sep. 15, 2018.

[63] Mixed integer linear programming benchmark (MIPLIB2010). [Online]. Available: http://plato.asu.edu/ftp/milpc.html. Accessed on: Sep. 15, 2018.

[64] AMPL. [Online]. Available: http://www.ampl.com. Accessed on: Sep. 15, 2018.

[65] GAMS. [Online]. Available: http://www.gams.com/. Accessed on: Sep. 15, 2018.

[66] AIMSS. [Online]. Available: http://www.aimms.com. Accessed on: Sep. 15, 2018.

[67] LINGO. [Online]. Available: http://www.lindo.com. Accessed on: Sep. 15, 2018.

[68] MPL. [Online]. Available: http://www.maximalsoftware.com. Accessed on: Sep. 15, 2018.

[69] TOMLAB. [Online]. Available: https://tomopt.com. Accessed on: Sep. 15, 2018.

[70] CMPL. [Online]. Available: https://projects.coin-or.org/Cmpl. Accessed on: Sep. 15, 2018.

[71] CVX. [Online]. Available: http://cvxr.com/cvx/. Accessed on: Sep. 15, 2018.

[72] Pyomo. [Online]. Available: http://www.pyomo.org/. Accessed on: Sep. 15, 2018.

[73] PuLP. [Online]. Available: https://pythonhosted.org/PuLP. Accessed on: Sep. 15, 2018.

[74] CyLP. [Online]. Available: https://github.com/coin-or/CyLP. Accessed on: Sep. 15, 2018.

[75] Yaposib. [Online]. Available: https://github.com/coin-or/yaposib. Accessed on: Sep. 15, 2018.

[76] FLOPC++. [Online]. Available: https://projects.coin-or.org/FlopC++. Accessed on: Sep. 15, 2018.

[77] JOM. [Online]. Available: http://www.net2plan.com/jom, https://github.com/girtel/JOM. Accessed on: Sep. 15, 2018.

[78] GUROBI. [Online]. Available: http://www.gurobi.com/. Accessed on: Sep. 15, 2018.

[79] CPLEX. [Online]. Available: https://www.ibm.com/analytics/cplex-optimizer. Accessed on: Sep. 15, 2018.

[80] XPRESS. [Online]. Available: http://www.fico.com/en/products/fico-xpress-solver. Accessed on: Sep. 15, 2018.

[81] MOSEK. [Online]. Available: https://www.mosek.com/products/mosek/. Accessed on: Sep. 15, 2018.

[82] J. D. Pintér, "LGO—A program system for continuous and lipschitz global optimization," in *Developments in Global Optimization*, I. M. Bomze, T. Csendes, R. Horst, and P. M. Pardalos, Eds. Boston, MA, USA: Springer, pp. 183–197, Accessed on: Sep. 15, 2018.

[83] KNITRO. [Online]. Available: https://www.artelys.com/en/optimization-tools/knitro. Accessed on: Sep. 15, 2018.

[84] SCIP. [Online]. Available: http://scip.zib.de/. Accessed on: Sep. 15, 2018.

[85] Google Optimization tools. [Online]. Available: https://developers.google.com/optimization/. Accessed on: Sep. 15, 2018.

[86] COIN-OR initiative. [Online]. Available: https://www.coin-or.org/. Accessed on: Sep. 15, 2018.

[87] GLPK. [Online]. Available: https://www.gnu.org/software/glpk/. Accessed on: Sep. 15, 2018.

[88] P. E. Gill, W. Murray, and M. A. Saunders, "SNOPT: An SQP algorithm for large-scale constrained optimization," *Soc. Ind. Appl. Math. Rev.*, vol. 47, no. 1, pp. 99–131, 2005.

[89] B. A. Murtagh and M. A. Saunders, "Large-scale linearly constrained optimization," *Math. Program.*, vol. 14, no. 1, pp. 41–72, 1978.

[90] LP_SOLVE. [Online]. Available: http://lpsolve.sourceforge.net/. Accessed on: Sep. 15, 2018.

[91] MIPCL. [Online]. Available: http://www.mipcl-cpp.appspot.com/. Accessed on: Sep. 15, 2018.

[92] NS-3. [Online]. Available: https://www.nsnam.org/. Accessed on: Sep. 15, 2018.

[93] OMNet++. [Online]. Available: https://omnetpp.org/. Accessed on: Sep. 15, 2018.

[94] Cisco WAE Design. [Online]. Available: https://www.cisco.com/c/en/us/td/docs/net_mgmt/wae/6-4/design/user/guide/WAE_Design_User_Guide.html. Accessed on: Sep. 15, 2018.

[95] Juniper IP/MPLS View. [Online]. Available: https://www.juniper.net/us/en/products-services/sdn/wandl/. Accessed on: Sep. 15, 2018.

[96] Packet Design. [Online]. Available: https://www.packetdesign.com/. Accessed on: Sep. 15, 2018.

[97] OPNET. [Online]. Available: https://www.riverbed.com/gb/products/steelcentral/opnet.html. Accessed on: Sep. 15, 2018.

[98] E-lighthouse Network Planner. [Online]. Available: http://e-lighthouse.com/. Accessed on: Sep. 15, 2018.

[99] Cisco optimization service for network function virtualization and virtual managed services. [Online]. Available: https://www.cisco.com/c/dam/en_us/about/doing_business/legal/service_descriptions/docs/cisco-optimization-service-for-network-function-virtualization-infrastructure.pdf. Accessed on: Sep. 15, 2018.

[100] Net2Plan repository. [Online]. Available: https://github.com/girtel/Net2Plan. Accessed on: Sep. 15, 2018.

[101] J. L. Izquierdo-Zaragoza *et al.*, "Maximizing IP fast rerouting coverage in survivable IP-over-WDM networks," in *Proc. Eur. Conf. Opt. Commun.*, Valencia, Spain, 2015, pp. 1–3.

[102] Net2Plan open API documentation. [Online]. Available: http://net2plan.com/documentation/current/javadoc/api/index.html. Accessed on: Sep. 15, 2018.

[103] P. Pavon-Marino *et al.*, "Evaluating internal blocking in noncontentionless flex-grid ROADMs [invited]," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 7, no. 3, pp. A474–A481, Mar. 2015.

[104] Net2Plan extension that uses the Optical Line Emulator (OLE) developed in TIP based on the GN model. [Online]. Available: https://github.com/girtel/Net2Plan-TIP-OpticalImpairments-OLE-Report. Accessed on: Sep. 15, 2018.

[105] J.-L. Izquierdo-Zaragoza *et al.*, "Leveraging Net2Plan planning tool for network orchestration in OpenDaylight," in *Proc. Int. Conf. Smart Commun. Netw. Technol.*, Vilanova i la Geltrú, Spain, 2014, pp. 1–6.

[106] J. L. Izquierdo-Zaragoza *et al.*, "Dynamic operation of an IP/MPLS-over-WDM network using an open-source active stateful BGP-LS-enabled multilayer PCE," in *Proc. Int. Conf. Transparent Opt. Netw.*, Trento, Italy, 2016, pp. 1–4.

[107] P. Sköldström, Ć. Rožić, and J.-J. Pedreno-Manresa, "Making powerful friends: Introducing ONOS and Net2Plan to each other," in *Proc. Int. Conf. Transparent Opt. Netw.*, Girona, Spain, 2017, pp. 1–4.

[108] Fluent OpenStack SDK for Java. [Online]. Available: http://www.openstack4j.com/. Accessed on: Sep. 15, 2018.

Authors' biographies not available at the time of publication.