Machine Learning-assisted Planning and Provisioning for SDN/NFV-enabled Metropolitan Networks

Sebastian Troia(1), David Eugui(2), Ignacio Martín(2), Ligia Maria Moreira Zorello(1), Guido Maier(1), José Alberto Hernández(2), Oscar González de Dios(3) , Miquel Garrich(4), José Luis Romero-Gázquez(4), Francisco-Javier Moreno-Muro(4), Pablo Pavón Mariño(4)(5), Ramon Casellas(6)

(1) *Politecnico di Milano, Dipartimento di Elettronica Informazione e Bioingegneria (DEIB), Milan, Italy*
(2) *University Carlos III de Madrid, Telematic Engineering Department, Madrid, Spain*
(3) *Telefonica Global CTO, Madrid, Spain*
(4) *Universidad Politécnica de Cartagena, Telecommunications Department, Cartagena, Spain*
(5) *E-lighthouse Network Solutions*
(6)*Centre Tecnològic de Telecomunicacions de Catalunya (CTTC/CERCA), Castelldefels, Barcelona, Spain*

***Abstract*—After more than ten years of research and development, Software-Defined Networking (SDN) and Network Function Virtualization (NFV) are finally going mainstream. The fifth generation telecommunication standard (5G) will make use of novel technologies to create increasingly intelligent and autonomous networks. The METRO-HAUL project proposes an advanced SDN/NFV metro-area infrastructure based on an optical backbone interconnecting edge-computing nodes, to support 5G and advanced services. In this work, we present the METRO-HAUL planning tool subsystem that aims to optimize network resources from two different perspectives: off-line network design and on-line resource allocation. Off-line network design algorithms are mainly devoted to capacity planning. Once network infrastructure is in production stages and operational, on-line resource allocation takes into account flows generated by end-user-oriented services that have different requirements in terms of bandwidth, delay, QoS and set of VNFs to be traversed. Through the paper, we describe the components inside the planning tool, which compose a framework that enables intelligent optimization algorithms based on Machine Learning (ML) to assist the control plane in taking strategic decisions. The proposed framework aims to guarantee a fair behavior towards past, current and future requests as network resource allocation decisions are assisted with ML approaches. Additionally, interaction schemes are proposed between the open-source JAVA-based Net2Plan tool, ML libraries and algorithms in Python easing algorithm development and prototyping for rapid interaction with SDN/NFV control and orchestration modules.**

Keywords—Software Defined Networking, Network Function Virtualization, Metro-Haul project, Machine Learning, Net2Plan, Network Optimization

I. Introduction

The aim of **Network Planning** is to optimize network resources from two different perspectives: off-line network design and on-line resource allocation. Off-line network design algorithms are mainly devoted to capacity planning both for green-field scenarios and partially deployed (*i.e.* brown field) deployments. Once network infrastructure is in production stages and operational, on-line resource allocation takes into account flows generated by end-user-oriented services that have different requirements in terms of bandwidth, delay and QoS. Therefore, the provision of each service requires the on-line solution of a specific resource-assignment problem (*reactive* operational mode of the control plane). On the other hand, there must be a coordinated control in the provision of system resources to the various requests, in order to guarantee a fair behavior towards past, current and future requests. For this reason, there is a need for intelligent optimization algorithms able to guarantee a fair treatment to all service requests. To achieve that, such algorithms suggest to the control plane when and how to re-optimize the network to improve resource utilization (*proactive* operational mode). Re-optimization should be computed by observing not just the present state, but also the past behavior and the future behavior by prediction. In brief, there is the need of a network planning and optimization system based on Machine Learning (ML), which assists the control plane in taking strategic decisions.

The Metro-Haul[1] project deals with an advanced metro-area infrastructure for Information Technology (IT) and networking, based on an optical backbone interconnecting edge-computing nodes, to support 5G and advanced services. In Metro-Haul's unified service platform architecture, a subsystem is dedicated to network planning and optimization, which takes into account the whole network infrastructure. A specific goal of the planning and optimization subsystem is to provide the *network resource* assignment to requests that come to the network either from the operator or as demands from vertical technologies / network users. Within the Metro-haul context, resources refer to: a) computing resources, w.r.t. deployment of virtual network functions in suitable virtual machines in the compute nodes; b) network resources, w.r.t. routing of connections or instantiation of traffic flows over the nodes and links of the multi-layer network infrastructure. Consequently, the planning tool needs to be interfaced to two main components of the network control plane: namely, the Software Defined Network (SDN) controller(s), responsible for the networking resources, and the Virtual Infrastructure Managers (VIM), responsible for the IT/Cloud resources. Moreover, the planning tool is interfaced to the Network Function Virtualization service Orchestrator (NFVO), in order to detect the service requests arriving at the network from the customers. In this paper, we present the Metro-Haul Planning Tool subsystem that aims to optimize and reconfigure network resources in an intelligent and proactive way. The paper is organized as follows: Sec. II is dedicated to related work. In Sec. III we describe its architecture and interface with the other

[1] Website: https://metro-haul.eu/

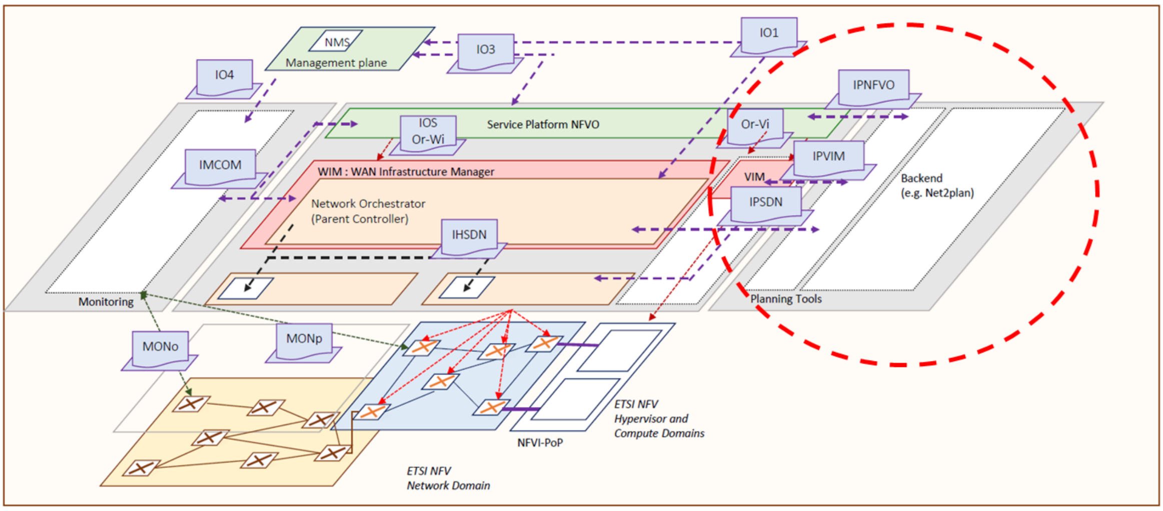

Fig. 1: Metro-Haul general architecture. The green blocks represent the management plane, which is composed of the service platform and the Network Management System (NMS), responsible for providing information about the service chain characteristics. The red blocks determine the infrastructure management systems, containing the network orchestrator and the virtual infrastructure manager. The monitoring module in the left side of the figure monitors the system to provide information that determines whether reconfiguration is needed. The planning tool module in the right is described in the following subsections. The remaining blocks (IPNFVO, IPVIM, IPSDN, IO1, Or-Vi, IO3, IO4, IOS Or-Wi, IMCOM, MONo and MONp) correspond to interfaces for information exchange between the modules. More information about the Metro-Haul architecture is depicted in [12][13].

control-plane components. In Sec IV we summarize and report the results of preliminary tests of the proposed subsystem. Finally, Sec. V concludes the paper.

II. Related Work

The rise of novel algorithms (*e.g.* Deep Learning and Deep Reinforcement Learning) in conjunction with the availability of techniques to analyze Big Data has risen a new wave of ML applications in computer networks [1], often focused on automated network management with reduced human intervention [2][3]. In particular, such novel techniques have shown applicability in network routing and virtualization [4][5]. However, ML algorithms need to be fed with large quality training datasets, which is sometimes difficult to obtain. Some studies have pointed in this direction reporting the lack of data in networking scenarios [6], the need for creating standardized datasets [2], the challenges arising from the collection of high-quality networking data [7] or showing the opportunity for upcoming technologies, such as SDN, to introduce network monitoring systems aware of the ML needs [8].

Concerning optical Wavelength Division Multiplexing (WDM) networks, the use of Artificial Intelligence and Machine Learning tools has the potential to provide solutions at the physical layer; such as to estimate optical signal quality of transmission, characterize and operate transmitters, mitigate nonlinearities at receivers, detect link failures or recommend wavelength assignments with low power excursion [9][10]. In the specific case of the network resources planning and optimization problems, Integer Linear Programming (ILP) formulations for its optimal solution exist since long ago [11]. In spite of the cost-minimized solutions provided by ILPs, they suffer from high computational complexity, often requiring minutes or hours to solve medium-size network topologies. A large number of Heuristic algorithms have also been proposed in the literature, offering faster-than-ILP solutions but rather sub-optimal.

In this work, we introduce a framework of network planning and optimization based both on classical ILP based algorithms and novel ones based on ML.

III. Metro-Haul Planning Tool

Fig. 1 shows the Metro-Haul Service Platform architecture, where the dashed red circle highlights the Placement, Planning, and Reconfiguration Subsystem. The main role of the network planner is to propose rearrangements of existing active service chains and network connections with the purpose of globally re-optimizing resource allocation to improve network performance. This implies to optimize resource allocation in optical metro networks and at the same time effectively provision Virtual Network Functions (VNFs) in specific computing nodes considering heterogeneous requirements. Re-optimization should not simply consider the current state, but in the light of the evolution of the network traffic in future periods, as it can be predicted by observing traffic for a given number of past epochs. Predictions are carried out by suitable algorithms based on machine learning. The proactive re-optimization service can be carried out periodically or it can be triggered by monitoring utilization of resources and detecting degradation above certain thresholds. Net2Plan[2] [14] has been chosen by the Metro-Haul consortium as the planning tool and common framework for collaboration among partners, exploiting its multiple functionalities for algorithm development, capacity planning, resource provisioning and automatic report generation. Net2Plan is a free and open-source Java tool developed, hosted and maintained by Girtel group at UPCT, devoted to planning, optimization, and

[2] Website: http://www.net2plan.com/

evaluation of communication networks. In the Metro-Haul context, Net2Plan is in charge of executing algorithms that perform both off-line network design and on-line resource allocation algorithms. This approach of implementing the planning and optimization service into a dedicated subsystem has several advantages that can be summarized as follows:

The service and network orchestrators – especially in case of open-source software – often do not have built-in optimization algorithms or, if they do, they commonly rely on well-known algorithms, not easily replaceable by others. By developing dedicated planning and optimization algorithms while interfacing with the controllers and orchestrators, full control is permitted to math and logic behind optimization routines. Thus, allowing upgrades and modifications enable the usage of more efficient algorithmic techniques in line with novel ML approaches.

The network operator has easy access to the algorithms and thus can customize the service by deciding his own policies and criteria. Since the Metro-Haul network is an experimental environment, it is important to have the possibility of easily testing new optimization and prediction algorithms.

Algorithms can be coded in Java and Python as well. One of the innovations of this planning module is to implement algorithms for reactive and proactive network optimization in different languages. Java is mostly used for algorithms based on linear programming, while Python, for ML-based algorithms thanks to the huge amount of open-source libraries available [15][16].

A. Network Planner Architecture

The Network Planner architecture is shown in Fig. 2. It is divided in two blocks. The Front-end block contains the interfaces that allow the exchange of information with the other components of Metro Haul control plane. The Back-end block contains the implementation of the computational modules necessary for the planning tool. The open-source Java-based Net2Plan [14] tool has been chosen as the Network Planner framework for placement, planning and reconfiguration of: VNFs, IT and network resources.

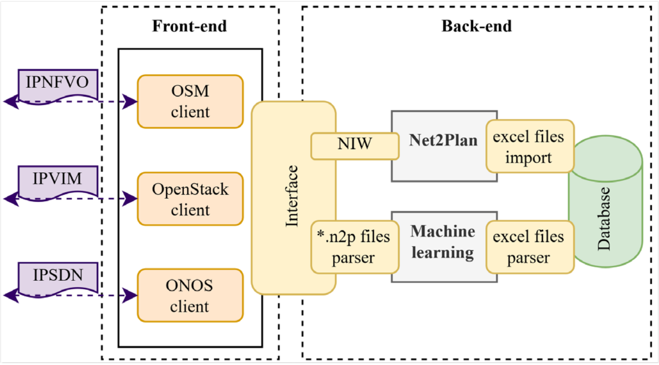

Fig. 2: Planning and reconfiguration system

1) Front-end

The Front-end includes a series of client modules to exchange information with the software components of the Control, Orchestration and Management (COM) architecture and consume it in the Network Planner framework. This information involves the network topology in terms of network nodes, links and traffic engineering attributes, such as occupied optical frequency slots or additive metrics as well as the status of the different hypervisors in computed nodes of the infrastructure. In particular, the exchange of information is performed exploiting three interfaces:

- Interface Planner – NFV-O (IPNFVO) aims to provision network services composed as an ordered sequence of VNFs through network service descriptors (NSD), which are the information model that represent network services in Open Source MANO (OSM)[3] terminology. OSM is an open-source NFV Management and Orchestration (MANO) implementation under the umbrella of ETSI, in charge of orchestrating the VNF placement and life cycle in one or more data centers, potentially connected by a transport network.
- Interface Planner – SDN controller(s) (IPSDN) aims to gather information from the transport infrastructure and provides transport network resource allocation in line with the requirements of the network services.

Relevant for IPNFVO and IPSDN, recent demonstrations reported the Planning Tool Net2Plan assisting an OSM instance in the optimal allocation and instantiation of network services (SCs) in a simulated transport network [15]. Subsequently, [17] demonstrates an ONOS controller emulating the packet-layer network and considered the end-to-end latency requirements to perform flow allocation jointly with the SC allocation and VNF instantiation via OSM.

- Interface Planner – VIM (IPVIM) interacts with the IT resource manager to enable multiple functionalities inherent of the NFV technology. In particular, a recent demonstration reported an open-source Net2Plan extension for interfacing multiple OpenStack instances, which enables multi-tenant slicing, IT resource visualization and VM migration [18].

Additionally, a REST-based interface has been recently proposed to enable third-party applications to perform Optimization as a Service [19].

2) Back-end

The Back-end consumes the information gathered from the Front-end to execute its algorithms with the eventual support/assistance of ML The Back-end block has been conceived as a very flexible environment which allows to easily plug-in a variety of computation algorithms implemented in heterogeneous software environments, provided they are compatible with the overall architecture. Specifically, the Back-end is currently composed of two modules. The first is populated by algorithms natively developed in the Net2Plan environment. The algorithms hosted in this module include: a) an NFV over IP over WDM (NIW) library [20][21], b) a VNF placement algorithm, c) a network Resource and Wavelength Allocation (RWA)

[3] Website: https://osm.etsi.org/

algorithm. a) is in an advanced stage of development and already included in the Net2Plan framework, b) and c) are currently under development and integration. All these algorithms adopt network optimization techniques, are oriented to connection and service-chain real-time provisioning and exchange information with the Front-end using NIW library integrated in Net2Plan. They are implemented in Java. The second module incorporates algorithms oriented to predictive periodical or off-line network re-optimization, they make use of ML techniques and are developed in Python.

The front and back end modules exchange information by means of native Net2Plan *.n2p files. They include: d) a ML-based RWA algorithm, e) a ML-based VNF-placement algorithm, f) a ML-based algorithm integrating VNF placement and RWA. d) is in advanced stage of development, e) will be developed in the next months, while the development of f) is a target for the end of the project. The Back-end module also includes a database to store any information consumed and produced by the algorithms. This permits the usage of historical data-sets that are particularly useful to train the ML algorithms. For the purpose of demonstration, a database solution based on MS Excel files was chosen, to maintain full compatibility with Net2Plan and speed-up the integration [22].

B. *Integration*

Fig. 3 shows the workflow involving the integrated components that summarizes the process to e.g. optimize network resources. It is divided into two different phases: training and real-time. In the training stage, the excel file containing the historical data regarding nodes, links, types of VNF executed, service chains and demands is provided to the back-end to train the machine learning algorithm. During the real-time phase, the interface between the front-end and back-end is responsible for providing the current network status via *.n2p files both to a simple use-case that is computed using NIW, and to the machine learning module, which calculates the optimal solution according to its training. The result including the topology and allocated demands is then transmitted to Net2Plan. One of the main advantages of the approach we have followed in defining the architecture of the Metro Haul Planning Tool is the possibility of having a wide and expandable library of algorithms that are ready to be tested in the MH network architecture. Therefore, the user has several degrees of freedom in choosing the algorithms that will be actually used for a specific demo or a specific application. Moreover, it will be possible to compare the results of different algorithms on the same problem, algorithms on the same problem, so to facilitate the evolution of the computation methods. For instance, we will be able to compare the single-step approach (integrated VNF placement + RWA) to the probably less performing but simpler two-steps (first VNF placement, then RWA).

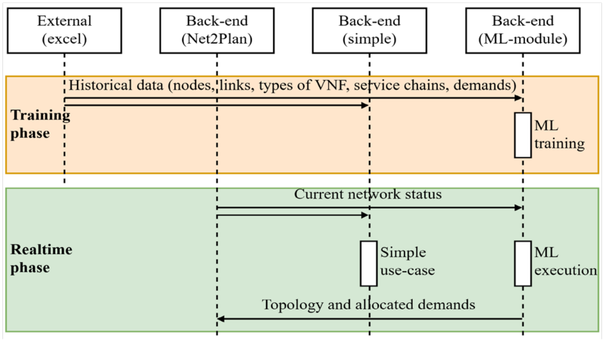

Fig. 3: Workflow showing the integrated components that shows both the training and real-time phases

C. *Network Planner Algorithmic View*

We consider 4 possible attributes of implemented algorithms, namely VNF Reactive, Network Reactive, VNF Proactive and Network Proactive, see Fig.4. A given algorithm, e.g. #4, performs functions associated to VNF and Network planning in a pro-active way. The algorithms provided by the partners of the project can be split into six different functionalities, according to the type of optimization they implement, *i.e.* they compute the configuration either in reactive or proactive modes in terms of VNFs, network resources or both.

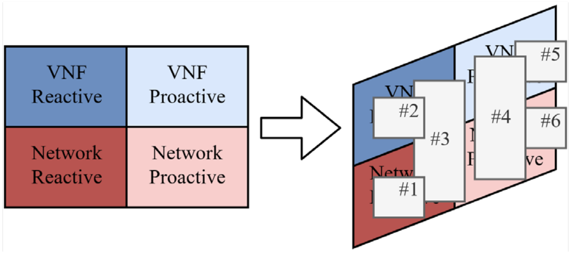

Fig. 4: Placement, planning and reconfiguration algorithmic view

IV. Preliminary Testing and Demos

The proposal described in this paper extends and unifies preliminary approaches developed in previous experiments and proof-of-concepts, mainly related to the back-end module and their implementation. Network planning in native Net2Plan environment is supported by the Java Optimization Modeler library (JOM) that assists Net2plan users in developing optimization problems to be solved using optimizers such as GLPK or CPLEX. In [19], the NIW library is presented in a tutorial manner with exemplary algorithms that contain different variations for the VNF instantiation and service-chain deployment procedures, while taking into account IP over WDM network resources. Then, those algorithms are applied on a reference topology illustrating the obtainable results. On the ML side, the initial successful experiment has been a demonstration in ECOC [23] of how ML-assisted routing could be integrated in an SDN-managed Mininet[4] network and the analysis of RWA as a ML problem solved through simple algorithms [24]. In the first work, we proposed the Machine Learning Routing Computation Module (MLRC), which combines ML-powered routing with an ONOS-controlled SDN network that measures traffic in the network and attempts online reconfiguration assisted by MLRC. We demonstrated how this module is correctly integrated into the SDN controller and how ML is able to quickly provide network reconfiguration upon traffic matrix alteration detections. The

[4] Website: http://mininet.org/

second approach addresses RWA problem from one question: Is ML able to solve RWA problems? The aim of the work was to identify whether machine learning can approximate an ILP solution to RWA at a much lower cost, especially in time. Using simple logistic regression over a synthetically generated set of matrices and different transponder configurations, we show that ML can be useful, offering time cuts of over 90%. In addition, this work shows how ML is capable of providing suitable and alternative solutions to the one given by the ILP, providing feasible network configurations in at least 65% of the time. Finally, a later work [23] combines these two approaches presenting an ML-based framework that can generate data, develop advanced ML models to emulate optimal ILP and similar algorithms for network problem resolution and their posterior implementation into an SDN network. Regarding ML-assisted RWA, we improve the algorithm in [24] by developing a Deep Neural Network classifier that clearly beats the previous approaches from logistic regression. In addition, we modified the behavior of the classifier to develop a ranking of most suitable routing and wavelength configurations instead of a single classification. This way, and using the neural network, the algorithm is capable of proposing suitable routing and wavelength configurations in at least 97% of cases and offer time cuts larger than 96% with respect to the ILP solution. Regarding ML-assisted IP routing and its SDN implementation, we refined the MLRC module and measured its reconfiguration time: MLRC is able to perform full reconfiguration, including the detection of changes in the traffic matrices, in approximately 80 ms. For reference, the ONOS Native Path Computation in the same network takes approximately 180 ms, which is much longer than the proposed solution.

V. Conclusion and Discussion

This paper has presented the Metro-Haul planning tool subsystem for optimizing and reconfiguring network resources in an intelligent and proactive way. In this work, we propose the introduction of different components to assist in network planning and SDN online re-configuration. For this purpose, the tool internals, including front-end and back-end components are described in detail along with their proposed integration. Furthermore, different existing and foreseen use cases are presented and analyzed. Previous works in the Scope of the Metro-Haulproject have shown that ML provides a clear benefit that can be leveraged for assisting in network configuration faster than solving ILP optimizations and more accurate than Heuristic solutions. Indeed, in [23][24] it has been shown how ML-assisted network configuration can yield efficient configuration proposals very fast that can be implemented into SDN networks. Hence, this proposal aims at the integration of ML-assisted enhancements into Net2Plan tool to facilitate network planning and SDN interfacing within the METRO-HAUL project scope.

Acknowledgment

The work leading to these results has been supported by the European Community under grant agreement no. 761727 Metro-Haul project.

References

[1] M. Wang, et al., “Machine learning for networking: Workflow, advances and opportunities”, IEEE Network, vol. 32, no. 2, pp 92–99, 2017.

[2] A. Mestres, et al., “Knowledge-defined networking”, ACM SIGCOMM Computer Communication Review, vol. 47, no. 3, pp. 2–10, 2017.

[3] J. Barron, et al., “Towards self-adaptive network management for a recursive network architecture”, in IEEE/IFIP Network Operations and Management Symposium (NOMS), pp. 1143–1148, 2016.

[4] S. Haeri, “Applications of reinforcement learning to routing and virtualization in computer networks”, Ph.D. dissertation, University of Malaysia, 2016.

[5] N. C. Luong, et al., “Applications of deep reinforcement learning in communications and networking: A survey”, 2018.

[6] M. Wang, et al., “Machine learning for networking: Workflow, advances and opportunities”, IEEE Network, vol. 32, no. 2, pp. 92–99, 2018.

[7] K. Zheng, et al., “Big data-driven optimization for mobile networks toward 5g”, IEEE Network, vol. 30, no. 1, pp. 44–51, 2016.

[8] I. F. Akyildiz, et al., “Research challenges for traffic engineering in software defined networks”, IEEE Network, vol. 30, no. 3, pp. 52–58, 2016.

[9] C. Rottondi, et al., “Machine-learning method for quality of transmission prediction of unestablished lightpaths”, IEEE/OSA Journal of Optical Communications and Networking, vol. 10, no. 2, pp. A286–A297, 2018.

[10] S. Shahkarami, et al., “Machine-learning-based soft-failure detection and identification in optical networks”, in 2018 Optical Fiber Communications Conference and Exposition (OFC), 2018.

[11] H. Zang, et al., “A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks”, Optical networks magazine, vol. 1, no. 1, pp. 47–60, 2000.

[12] METRO-HAUL Control and Management Requirements and Framework. https://metro-haul.eu/deliverables/

[13] R. Casellas, et al., “Metro-haul: SDN control and orchestration of disaggregated optical networks with model-driven development”, in International Conference on Transparent Optical Networks, 2018.

[14] P. Pavon-Marino, et al., "Net2plan: an open source network planning tool for bridging the gap between academia and industry", IEEE Network, vol. 29, no. 5, pp 90–96, 2015.

[15] M. Abadi et al. "Tensorflow: A system for large-scale machine learning", in USENIX Symposium on Operating Systems Design and Implementation (OSDI’16), 2016.

[16] J. Bergstra, et al., "Theano: a CPU and GPU math expression compiler", in Python for scientific computing conference (SciPy), vol. 4, no. 3, 2010.

[17] F. J. Moreno-Muro, et al., “Latency-aware optimization of service chain allocation with joint VNF instantiation and SDN network control”, in European Conference on Optical Communication, 2018.

[18] M. Garrich, et al., “The Net2Plan-OpenStack Project: IT Resource Manager for Metropolitan SDN/NFV Ecosystems”, in Optical Networking and Communication Conference and Exhibition, 2019.

[19] M. Garrich, et al. “The Net2Plan-based Library for NFV over IP over WDM Networks”, in International Conference on Transparent Optical Networks, 2019.

[20] NFV over IP over WDM (NIW) library. [Online]. Available. https://github.com/girtel/Net2Plan/tree/master/Net2Plan-Core/src/main/java/com/net2plan/research/niw/networkModel

[21] NIW Javadoc documentation. [Online]. Available. http://www.net2plan.com/documentation/current/javadoc/api/com/net2plan/research/niw/networkModel/package-frame.html.

[22] Excel file template of the NIW library. [Online]. Available. https://github.com/girtel/MetroHaulPlanner/blob/master/MetroNetwork_Template.xlsx.

[23] S. Troia, et al., “Machine-Learning-Assisted Routing in SDN-based Optical Networks”, in European Conference on Optical Communications (ECOC'18), 2018.

[24] I. Martin, et al., “Is Machine Learning Suitable for Solving RWA Problems in Optical Networks?”, in European Conference on Optical Communications (ECOC'18), 2018.