Fichera, S., Martínez, R., Martini, B., Gharbaoui, M., Casellas, R., Vilalta, R., Muñoz, R. y Castoldi, P. (2018). Latency-Aware Network Service Orchestration over an SDN-Controlled Multi-Layer Transport Infrastructure. In: *20th International Conference on Transparent Optical Networks (ICTON),* pp.1-4.

DOI: 10.1109/ICTON.2018.8473817

Latency-aware Network Service Orchestration over an SDN-controlled Multi-Layer Transport Infrastructure

S. Fichera*, R. Martinez[§], B. Martini[†], M. Gharbaoui[†], R. Casellas[§], R. Vilalta[§], R. Muñoz[§], P. Castoldi*
[†] *National Inter-University Consortium for Telecommunications (CNIT), Pisa, Italy*
Tel: +39 050 549 2245, Fax: +39 050 549 2250, e-mail: barbara.martini@cnit.it
** Scuola Superiore Sant'Anna, Pisa, Italy*
§Centre Tecnològic de Telecomunicacions de Catalunya (CTTC/CERCA), Barcelona (Spain)

ABSTRACT

In this paper, we present latency-aware orchestration strategies that jointly consider satisfying both the allocation of computing resources (in distributed DCs) and the bandwidth and latency networks requirements, which are experimentally evaluated within a Multi-Layer (Packet over Optical Flexi-Grid) Transport Network and considering different DC set-ups and capabilities.

Index Terms—Orchestration, 5G, SDN, NFV, VNF Forwarding Graph, Virtual Network Function

1. INTRODUCTION

Through network slicing, upcoming 5G network services will be delivered and tailored to serve many different application scenarios (e.g., automotive, smart manufacturing, cloud robotics) that will demand different network functions with various levels of QoS (bandwidth, latency, etc.). Indeed, 5G is not only accounted to as a new radio technology. It also refers to a new way to deploy networks thanks to novel capabilities offered by both Network Function Virtualization (NFV) and Software-Defined Networking (SDN), i.e., *softwarization* [1]. NFV promotes a scenario in which network functions (i.e., from a switch/router to a software middle-box) are deployed as virtual machines, i.e., Virtual Network Functions (VNFs), either centralized in the cloud or distributed in clusters of small- and medium-DCs located at the edge of the network [2][3]. The topological terms of such a distributed VNF deployment are specified by the VNF Forwarding Graphs (VNF-FGs) [4]. On the other hand, SDN provides programming abstractions that can be effectively exploited for the dynamic enforcement and in-line steering rules of data flows across VNFs taking part in the VNF-FG [5][6].

The possibility to dynamically provision network services in the cloud is attracting the interest of the network and service operators as well as standardization organizations [4] eager to leverage the high flexibility, rapidity and cost-effectiveness of the network service provisioning offered by the *softwarization* approach [4][7][8]. However, the above *softwarization* capabilities are attained at the expenses of imposing a burden on the DCs and on the (metro-core) network interconnecting the DCs. Moreover, the VNFs may experience additional latency caused by the delocalization of the involved VNFs as well as by the limited DC resource availability, especially at the network edge. Thus, to exploit *softwarization* potentialities, it is crucial to conveniently select the DCs hosting the VNFs considering the current DC and network resource availability while ensuring that both required capacity and delay performance by 5G applications is guaranteed [9][10].

Toward this direction, we investigate the dynamic VNF-FG allocation over distributed DCs interconnected through a Multi-Layer Network (MLN), i.e., Packet over Optical Flexi-Grid transport infrastructure. VNF-FG allocation requests are handled by a Cloud/Network Orchestrator composed of two main entities: the *Allocator* for processing VNF-FG requests and performing DC resource allocations; and a the Transport SDN controller to compute/configure MLN connections based on an the abstract network view including the topology and network resource attributes (link bandwidth, latency, cost). Specifically, this work presents two on-line orchestration strategies for the joint selection of DC and network resources aiming at addressing latency requirements for a given network service (VNF-FG) request while attaining an efficient use of the resources (mostly networking). The devised Cloud/Network Orchestrator and both the orchestration strategies are experimentally evaluated within the CTTC ADRENALINE[11] testbed to show practical feasibility of the proposed orchestration strategies and highlight their implementation implications on top of a MLN.

2. NETWORK SERVICE ORCHESTRATION IN MULTILAYER NETWORKS

This paper considers a Cloud/Network Orchestrator dynamically and automatically deploying VNF-FG requests by making decisions on where to allocate VNFs and on how inter-connect them based on the capabilities and current availability of (i) computing resources (i.e., CPU, storage and memory) of a distributed DC infrastructure, and (ii) network resources (i.e., links capacity) of the underlying transport infrastructure. The orchestration decisions are made based on different strategies depending on management targets, e.g., resource consolidation, or specified service requirements, e.g., minimum service latency.

Fig.1 (top) shows the two main building blocks of the Cloud/Network Orchestrator: the *Allocator* and the T-SDN controller. The *Allocator* receives, computes, and dynamically accommodates VNF-FG requests (*req*) based on the demanded requirements. The network service request conveys the VNF-FG detailing a graph with vertices (i.e., VNFs) featured by the amount of required computing resources (i.e., number of Virtual Machines (VMs), CPU, RAM and storage per VM), and with edges (i.e., virtual links connecting VNFs) featured by the packet inter-DC bandwidth demand (Bw, b/s). Latency requirements could be also specified in terms of the network delay experienced by the data while they are transmitted from a VNF to another VNF (i.e., L, ms). With respect to the ETSI orchestration framework, the *Allocator* plays the role of the Resource Orchestration functions of the NFV Orchestrator [4]. Indeed, the *Allocator* decides the allocation of a set of virtual resources (i.e., VMs and virtual links). This is done based on the resource capacity and availability offered by both DCs (stored in a repository called Cloud Database, DB) and the (abstracted) interconnection network view kept in a Traffic Engineering Database (TED). Hence, upon receiving a new *req,* the *Allocator* first checks whether the *req* can be accommodated with regard to the DC resources. Next, it interfaces the T-SDN controller (depending on the adopted approach) to retrieve (abstract) network information, to request inter-DC path computations and to instantiate inter-DC connections. The T-SDN Controller (based on a PCE Central Controller implementation) supports the following functions: processing packet connection requests arriving from the *Allocator* (via a NBI API that uses the PCEP protocol), performing MLN path computations using an updated view provided by the TED repository, record of the active packet and optical connections within the LSP Database (LSPDB), and configuring the computed packet and optical network elements (i.e., switches and Sliceable Bandwidth Variable Transponders -SBVTs).

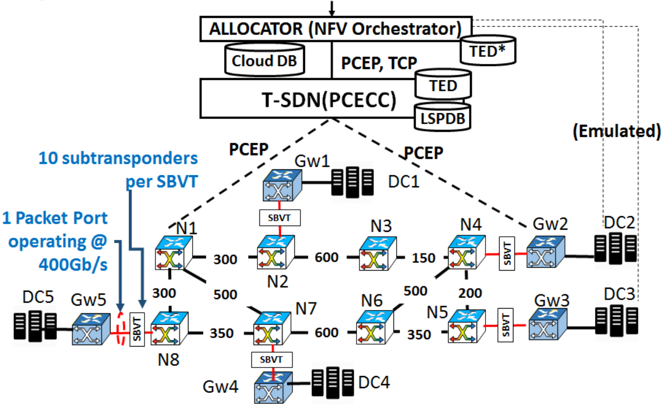

Fig. 1 Network scenario and Cloud/Network Orchestrator architecture

Fig.1 (bottom) shows the reference network scenario which is a transport MLN interconnecting DCs. Particularly, the MLN combines both packet and flexi-grid optical technologies leveraging the effective packet traffic grooming and the optical spectral efficiently and flexibility. Thus, a computed MLN path may be composed of re-using the spare capacity of existing virtual links (VLs) and/or allocating new flexi-grid optical path segment. For the sake of clarification, once an optical flexi-grid path is set up, this becomes as a packet VL between the optical path endpoints. The VL inherits some of the traffic engineering attributes of the underlying optical path such as the end-to-end delay, cost, unreserved bandwidth. Additionally, due to the mismatch between the connectivity demands and the optical transport capacity, in a MLN it is very likely (and desirable) that a number of disregarded packet connections can be routed over the same optical path (i.e., VL) leading to improve the overall network resource utilization in terms of SBVT, optical spectrum and packet ports.

Each (emulated) DC is assumed to be connected to a packet node (Gateway, Gw). Each Gw is in turn connected to the flexi-grid optical network via SBVT. MLN path computation mechanisms for computing packet inter-DC paths relies on the possession of an abstract network information (ANI) at the *Allocator*. That is, the *Allocator* retrieves from the T-SDN Controller (through a TCP protocol) abstracted network information based on active packet VLs and their characteristics (i.e., Gw endpoints, cost, available bandwidth and latency). By doing so, when the *Allocator* has selected the DCs to host the VNFs, it checks whether a packet path using the existing VLs is feasible for VNF inter-connection. If a path is feasible, then it is instantiated; otherwise (i.e., in case of lack of connectivity in the packet VLs), the *Allocator* must rely on the T-SDN controller for computing and providing the targeted VNF connectivity.

3. ON-LINE LATENCY-AWARE RESOURCE ORCHESTRATION

Two on-line orchestration strategies are presented for the joint selection of DC and network resources aiming at addressing latency requirements while attaining an efficient use of the resources (mostly networking) for a given *req*. Below we present the orchestration strategies by considering that a VNF-FG request is composed of two VNFs to be deployed into as many DCs (i.e., *src* DC and *dst* DC) to be selected:

- *Minimum-Distance* (MD) strategy aims at minimizing the propagation latency experienced by the data while traversing DC interconnections. More precisely, given an *src* DC, this strategy seeks for the *dst* DC with the shortest distance (in km) from the *src* DC provided that it has sufficient available computing resources (i.e., CPU, RAM, storage), and inter-DC interconnection has enough available bandwidth to satisfy the *req*.
- *Latency-Constrained* (LC) strategy aims at addressing latency constraints without putting any restrictions to the location of *dst* DC. Thus, the selection of the *dst* DC is not constrained by the network distance from the *src* DC. Instead, the *dst* DC can be either of other DCs put into operation and connected to the network provided that the network path connecting *src* and *dst* DCs addresses specified latency requirements (in

addition to bandwidth demand). In this case, the selection of *dst* DC can be based on cloud-centric information, e.g., current load of DC servers, and provided by a cloud management infrastructure system driven by a specified management target, e.g., load balancing.

The workflows related to the two strategies are shown in Fig.2. On the right-hand side, the workflow related to the MD strategy is reported. An incoming *req,* clarified in the figure, is received and processed by the *Allocator* containing: (*i*) the *src* DC and the number of VMs to be deployed, (*ii*) the number of VMs to be deployed in the *dst* DC, (*iii*) the CPU, memory and disk demand for each VM, and (*iv*) the bandwidth demand for the *src* and *dst* DC inter-connection. The *Allocator* checks first whether the requested amount of computing resources can be occupied in the *src* DC as well as there is at least one candidate *dst* DC with sufficient available IT resources. If either condition fails, the *req* is blocked and the previously allocated computing resources are released. Otherwise, for each candidate *dst* DC from a potential set of *N*, the *Allocator* asks to the T-SDN controller to compute a feasible path within the MLN infrastructure. This is done sending a *PCEP PCReq* message. This message carries the source and destination packet nodes (*Gw*) attached to the *src* and candidate *dst* DCs along with the targeted bandwidth (*Bw)*. Upon receiving the *PCEP PCReq* message at the T-SDN controller, the MLN routing algorithm described in [12] is triggered. Such an algorithm outputs a feasible shortest MLN path within a packet over flexi-grid network where the objective function is to attain the most efficient use of the network resources in terms of the SBVT's subtransponders, optical spectrum and packet ports. If a feasible path is found from the *src* to the *dst Gws*, a *PCEP PCRep* message is returned to the *Allocator* with the whole computed route (i.e., nodes, links, selected frequency slot and modulation format, and number of SBVT's subtransponders) along with a metric reflecting the actual distance (in km) between the two endpoints. Next, the *Allocator* selects the *dst* DC having the shortest metric (i.e., distance). Afterwards, the *Allocator* addresses the actual allocation of the computing resources at the selected *dst* DC as well as the network resources over the pre-computed path to enable the inter-DC connectivity. For the latter, as shown in the workflow, the *Allocator* relies on the abstracted network view provided by aforementioned ANI model. That is, the *Allocator* first seeks for a route between the selected *src* and *ds*t *Gws* reusing the spare capacity within existing packet VLs. If this succeeds, the allocation of the demanded *Bw* at the computed VLs is done by the T-SDN controller (via a *PCEP PCInitiate* message). This message includes information about the *Bw* and the selected VLs (Explicit Route Object – ERO – contents). Otherwise, the *Allocator* cannot find a feasible route based on the ANI view and thus, delegates to the T-SDN controller the MLN route computation and its subsequent provisioning. In both options, when a connection succeeds, a *PCEP PCRpt* message is sent back to the *Allocator* confirming that the *req* is established.

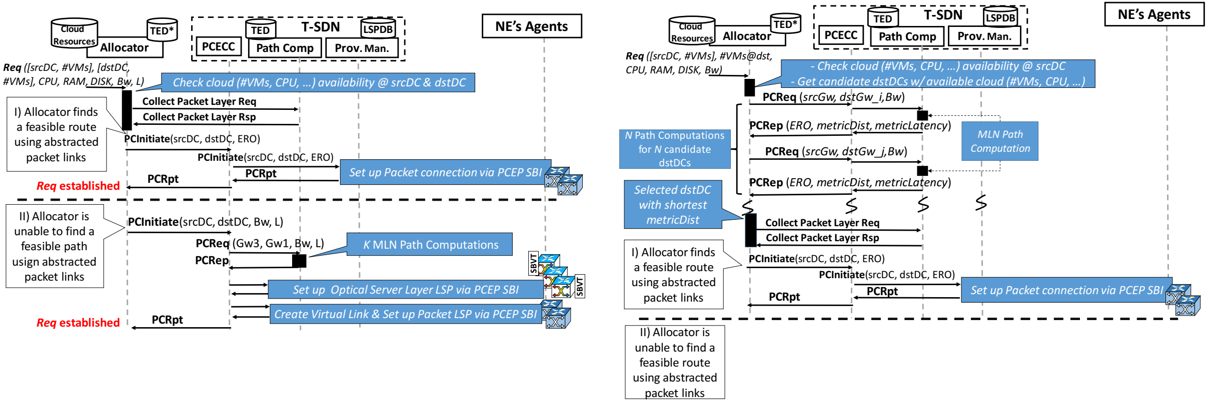

Fig. 2 Allocator - T-SDN Controller Workflows: Latency-Constrained Approach (left); Minimum-Distance Approach (right)

On the left-hand side of Fig.2, the workflow for the LC strategy is shown. The incoming *req,* different from the previous one, is received and processed by the *Allocator*. The *req* specifies the following parameters: (*i*) the *src* DC and the number of VMs to be deployed, (*ii*) the *dst* DC and the number of VMs to be deployed, (*iii*) the CPU, memory and disk demand for each VM, (*iv*) the bandwidth demand for the *src* and *dst* DC interconnection, and (*v*) the maximum allowed network latency. The *Allocator* firstly checks whether sufficient computing resources can be accommodated at both the *src* and *dst* DCs. If one of them does not have sufficient computing resources, the *req* is blocked. Conversely, as in MD, the *Allocator* checks whether, relying on the ANI information, a feasible packet path using VLs satisfies the *bw* and *L* requirements. The objective is again fostering grooming strategies over the existing packet VLs. If a feasible path is found, this is instantiated as above sending the *PCEP PCInitiate* message to the T-SDN controller. If not, the *Allocator* delegates to the he T-SDN controller computing a MLN path dealing with the *bw* and *L* constraints. In this case, the T-SDN Controller computes a K-CSPF path ordered by their cost and, in case of the same cost, they are sorted by the lowest latency. For the resulting MLN path, first it is needed to allocate the computed server layer (optical resources), which eventually will derive on packet VLs enabling the inter-DC packet connection. More details about the PCEP control messages within the T-SDN controller are found in [12].

4. EPERIMENTAL VALIDATION

The experimental performance evaluation of the proposed orchestration strategies has been carried out within the CTTC ADRENALINE testbed rolling out a cloud/network infrastructure as shown in Fig. 1. This infrastructure is formed by 5 (emulated) DCs linked to 5 Gw nodes. Each *Gw* node has a single port (operating at 400Gb/s) and is connected to the optical flexi-grid network via an SBVT with 10 subtransponders. Each subtransponder supports 3 different modulation formats (MFs) (i.e., DP-16QAM, DP-8QAM and DP-QPSK) enabling 3 respective bit rates (200, 150 and 100 Gb/s) for different maximum distances (650, 1000 and 3000 km). Optical links support 128 Nominal Central Frequencies spaced 6.25GHz. Optical fiber distances, indicated in Fig.1, are used for exploring different MFs during the K-CSPF computation and determines the accumulated propagation path delay needed for checking the latency restriction. A distributed DC environment has been emulated with a composition of *small* DC (supposed to run at the edge of the network to exploit user proximity), *large* DC (supposed to run beyond core network as traditional cloud DC) and *medium* DC (supposed to run across a metro network). Each DC size features different computing capabilities: 100 CPU cores, 256 GB for RAM and Storage for each small DCs and 5 (10) times the capacity of a *small* DC for the *medium* (*large*) DCs [13]. *Small* DCs are connected to *Gw2* and *Gw3*, *medium* DCs are connected to *Gw1* and *Gw4*, and the *large* DC is connected to *Gw5*. The experimental evaluation is performed sending out 1000 requests to the *Allocator* generated according to a Poisson distribution with a mean inter-arrival time of 25s and an exponential duration (service time) of mean varied between 50, 75, 100 and 150 seconds. The following requests' parameters are randomly generated: the number of VMs per DC is uniformly distributed [1, 5] while the computing demands for CPU, RAM and Storage in the range [1, 19]; the *Bw* is randomly selected among [10, 40, 100] Gbps and the *L* ranges [6, 12] ms.
The following figures compare the obtained results for both approaches.

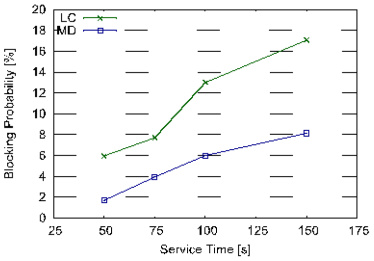

Fig.3 Overall Blocking Probability

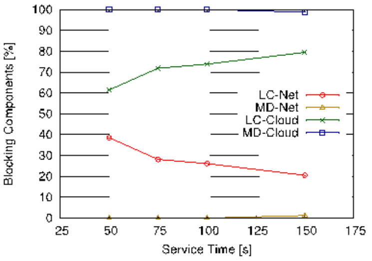

Fig. 4 Blocking Components Percentage

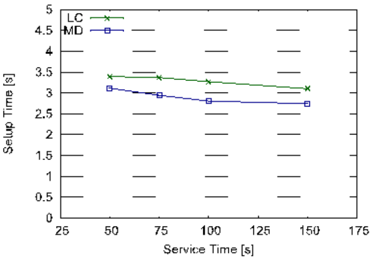

Fig. 5 Paths Setup Time vs. Service Time

Fig.3 shows the overall blocking probability. We notice that LC approach leads to block more requests than MD. Observing Fig. 4, we can notice that in the LC the main blocking cause is due to the network computation failure since indeed LC should addresses not only *Bw* constraint (as in MD) but also the *L* requirement. The network blocking significance in LC decreases when increasing the mean service time where the compute resources blocking becomes more relevant since such resources tend to be occupied for more time. Computing resources are the first resources being sought, so if the *Allocator* cannot accommodate them, the *req* is directly rejected without verifying the network availability. Fig.5 shows the average setup time of the established *reqs*. The MD needs less time to set up a *req* because given the selection of the closest *dst* DC for any *src* DC it is likely to have the same couple of selected DCs (i.e., *src DC*, *dst DC*) and an already active VL that connects them. Therefore, if we have enough bandwidth resources, the *Allocator* can accommodate the new *req* by using the already active VL(s) and, in practice, without requiring T-SDN controller connectivity computation. This behavior is clearly reflected into the total number of created VLs (Fig. 6). In this regard, observe that MD creates less VLs compared to LC. This performance metric also allows indicating how the adopted strategy leads to actually attain an efficient use of the overall network resources. That is, less VLs being created means allocating less optical resources, so the chance to accommodate more *reqs* in terms of network availability is higher. This is also the reason why, in Fig.4, for the MD the network blocking is almost equal to 0.

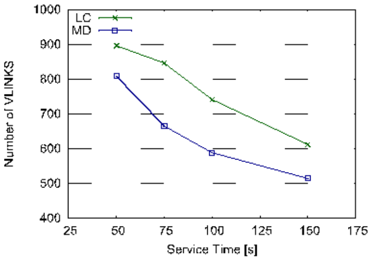

Fig.6 Number of VLINKS

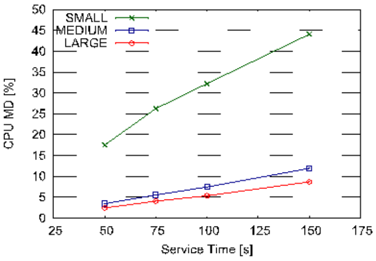

Fig. 7 Percentage of CPU Usage – MD case

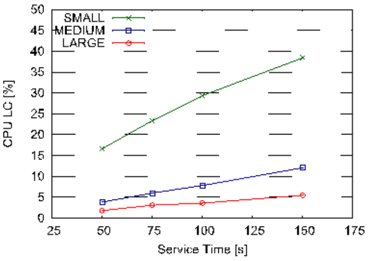

Fig. 8 Percentage of CPU Usage – LC case

In Fig. 7 and Fig.8, we show the comparison between the average CPU utilization for each dimension of DC. As expected, the more occupied DCs are in both cases the small ones, because they have less resources. We can notice that with a service time of 150s the average consumption of CPU reaches the 45% in the MD, whereas in the LC it is below 40%. This behavior is due to where the DCs are placed within the transport network according to their dimension. In fact, in the MD approach, each time that the source is a *small* DC, if enough computing resources are available, the selected *dst* DC will be the other small because of its shortest distance. In the LC approach, the couple (*src* DC, *dst* DC) is not constrained by their reciprocal position and, thus their selections are more spread and, consequently, all DCs are less loaded with respect to the MD.

5. CONCLUSIONS

In this paper, we have presented a Cloud/Network orchestrator operating on top of a transport MLN implementing two orchestration strategies aiming at addressing specified latency requirements. We reported the performance results of both devised strategies experimentally obtained within the CTTC ADRENALINE testbed. The results show the different impact of the proposed strategies in terms of the resource consumption and, thus, of the *reqs'* success rate. The attained results allow highlighting the implications of implementing the orchestration strategies on top of a real experimental platform using different patterns of the required layer interactions also affecting the average setup time.

ACKNOWLEDGMENTS

This work is partially funded by the EU H2020 5G TRANSFORMER project (761536).

REFERENCES

[1] IEEE SDN Blog, http://ieee-sdn.blogspot.it/

[2] A. Manzalini, R. Saracco, “Software Networks at the Edge: a shift of paradigm” in Proc. *IEEE Workshop SDN4FNS*, Nov 2013.

[3] H. Hawilo, A. Shami, M. Mirahmadi, R. Asal, “NFV: State of the Art, Challenges and Implementation in Next Generation Mobile Networks (vEPC),” *IEEE Network*, vol. 28, no. 6, pp. 18-26, Nov.-Dec. 2014.

[4] ETSI, GS NFV-MAN 001, “Network functions virtualisation (NFV) management and orchestration”

[5] A. Hakiri, P. Berthou, “Leveraging SDN for The 5G Networks: Trends, Prospects and Challenges,” arXiv preprint:1506.02876, 2015.

[6] B. Martini, F. Paganelli, A. A. Mohammed, M. Gharbaoui, A. Sgambelluri and P. Castoldi, "SDN controller for context-aware data delivery in dynamic service chaining,"in Proc of IEEE 2015 NetSoft, London, 2015, pp. 1-5.

[7] “Mobile edge computing - introductory technical white paper,” Tech. Rep., 2014.

[8] A. Rostami, et. al., “First experimental demonstration of orchestration of optical transport, RAN and cloud based on SDN,” in Proc. *Optical Fiber Communications Conference and Exhibition* (OFC), March 2015, pp. 1–3.

[9] D. Adami, B. Martini, M. Gharbaoui, P. Castoldi, G. Antichi, and S. Giordano, “Effective resource control strategies using openflow in cloud data center,” in Proc. *IFIP/IEEE International Sympsium on Integrated Network Management* (IM), 2013.

[10] B. Martini, et. al., “Latency-aware composition of virtual functions in 5G,” in Proc.of *IEEE* NetSoft, pp. 1–6, April 2015.

[11] R. Muñoz, et. al., “The ADRENALINE Testbed: An SDN/NFV Packet/Optical Transport Network and Edge/Core Cloud Platform for End-to-End 5G and IoT Services”, in Proc. of EUCNC 2017.

[12] R. Martínez, et. al., “Experimental Evaluation of a PCE Transport SDN Controller for Dynamic Grooming in Packet Optical over Flexi-Grid Optical Networks”, in Proc. of *ECOC 2017*, Sept. 2017.

[13] X. Wang, et. al., “Five resource allocation strategies for the cloud infrastructure,” in Proc. of ONDM, 2013.