

Received August 6, 2021, accepted September 6, 2021, date of publication September 23, 2021, date of current version October 4, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3115173

Resource Orchestration Strategies With Retrials for Latency-Sensitive Network Slicing Over Distributed Telco Clouds

M. GHARBAOUI¹, B. MARTINI², G. CECCHETTI¹, AND P. CASTOLDI¹, (Senior Member, IEEE)

¹TeCIP Institute, Scuola Superiore Sant'Anna, 56127 Pisa, Italy

²Photonic Networks and Technologies Laboratory, National Inter-University Consortium for Telecommunications (CNIT), 56124 Pisa, Italy

Corresponding author: M. Gharbaoui (molka.gharbaoui@santannapisa.it)

This work was supported in part by the Department of Excellence in Robotics and Artificial Intelligence by Ministero dell'Istruzione, dell'Università e della Ricerca (MIUR) to Scuola Superiore Sant'Anna, and in part by the Project 5GROWTH under Agreement 856709.

ABSTRACT The new radio technologies (i.e. 5G and beyond) will allow a new generation of innovative services operated by vertical industries (e.g. robotic cloud, autonomous vehicles, etc.) with more stringent QoS requirements, especially in terms of end-to-end latency. Other technological changes, such as Network Function Virtualization (NFV) and Software-Defined Networking (SDN), will bring unique service capabilities to networks by enabling flexible network slicing that can be tailored to the needs of vertical services. However, effective orchestration strategies need to be put in place to offer latency minimization while also maximizing resource utilization for telco providers to address vertical requirements and increase their revenue. Looking at this objective, this paper addresses a latency-sensitive orchestration problem by proposing different strategies for the coordinated selection of virtual resources (network, computational, and storage resources) in distributed DCs while meeting vertical requirements (e.g., bandwidth demand) for network slicing. Three orchestration strategies are presented to minimize latency or the blocking probability through effective resource utilization. To further reduce the slice request blocking, orchestration strategies also encompass a retrial mechanism applied to rejected slice requests. Regarding latency, two components were considered, namely processing and network latency. An extensive set of simulations was carried out over a wide and composite telco cloud infrastructure in which different types of data centers coexist characterized by a different network location, size, and processing capacity. The results compare the behavior of the strategies in addressing latency minimization and service request fulfillment, also considering the impact of the retrial mechanism.

INDEX TERMS Cloud, distributed data centers, NFV, retrial, SDN, virtual network function.

I. INTRODUCTION

The new radio capabilities of 5G and beyond networks will significantly foster higher communication performance that will enable a new breed of applications in several vertical domains (e.g., multimedia, cloud robotics, autonomous vehicles) featuring stringent yet diverse QoS requirements [1], [2]. For instance, IP multimedia services (e.g., VoIP) demand medium bandwidth, low delays and high CPU requirements; content delivery applications may ask for medium bandwidth, no relevant delay bounds and low CPU requirements (in case of video on demand services) or for

high bandwidth requirements, stringent delay bounds and medium CPU demands (in case of live streaming) [3].

A parallel and equally relevant technological shift in 5G networks is *network softwarization* that exploits software-based technologies (i.e., Network Function Virtualization (NFV) and Software-Defined Networking (SDN)) to foster increased automation and flexibility in network service deployments [4], [5]. Network functions will be set-up as virtual machines, i.e., Virtualized Network Functions (VNFs) in distributed telco Data Centers (DCs), i.e., telco clouds, and dynamically and programmatically interconnected via software network controllers. A scenario can be envisioned where telco service providers will be able to elastically provision partitions out of their virtualized infrastructure (i.e., *network slices*) tailored to the specific needs of vertical

The associate editor coordinating the review of this manuscript and approving it for publication was Muhamamd Aleem.

applications to address different scenarios or user requirements. A correlated trend is the *Mobile Edge Computing*, which fosters the deployment of distributed micro-clouds at the network edge (e.g., base stations, telco Point of Presence (PoP)) to run service and network functions closer to the user, i.e., service platforms that need a tighter interaction with users (e.g., content caches, video rendering), network functions properly scaled to assure the quality of experience expected by final users (e.g., user plane packet gateway functions, traffic optimization) [6].

This distributed and decentralized cloud-based 5G network architecture is posing new challenges to (telco) service providers to fully exploit the potentialities offered by *network slicing*, especially in terms of resource orchestration [7]. In particular, the prominent problem is how to dynamically and coordinately select and allot both (virtual) computational and network resources in telco clouds while effectively addressing diverse vertical QoS requirements in terms of end-to-end latency performance, processing capacity and bandwidth demands [8]. This is both a network and service management problem for telco service providers willing to increase the utilization rate of their (virtual) resources to optimize costs as well as fulfill as many vertical slice requests as possible to increase revenues.

In the light of above considerations, this work addresses the resource orchestration problem while allocating VNFs to establish network slices (i.e., VNF placement problem in a distributed telco cloud infrastructure with an eye to meet both resource management and service delivery targets of telco service providers and minimize end-to-end latency performance for the benefit of vertical applications. More specifically, we analyse a set of orchestration strategies featuring a coordinated selection of both computational and network resources to set-up slices while addressing (i) vertical requirements in terms of computational (e.g., CPU) demand and network (e.g., bandwidth) demand, (ii) effective usage of resources across DCs and the interconnection network, (iii) higher acceptance rate of the slice service requests through attempting a retrial of slice requests processing in case of lack of available (virtual) resources, and (iv) minimization of end-to-end latency performance offered by established slices to 5G vertical applications.

We evaluated the devised strategies based on a comprehensive set of simulations and we identified different performance indicators (e.g., blocking probability of slice requests, end-to-end latency performance) to thoroughly compare the performance offered by each of the strategies in a dynamic scenario of VNFs request generation. Performance are also evaluated when a retrial mechanism is adopted while assessing its benefits in terms of reduced blocking probability as well as its impact in terms of increasing network slice deployment time. We considered as reference scenario a composite yet realistic 5G infrastructure composed of geographically-distributed clouds (i.e., DCs) with different size and capacity (e.g., large cloud DCs in combination with resource-constrained *edge DCs*) and interconnected through

a wide area network spanning from the access/metro segment to the core network. As a representative of such an interconnection network, a flexi-grid optical wide-area network is considered in our evaluation thanks to the flexibility offered by elastic and dynamic allocation of optical link bandwidth [9].

The novel aspects of this work lay in discussing different orchestration strategies as representatives of different categories of policies to support service providers with design guidelines to fulfill network slice requirements from verticals (especially in terms of end-to-end latency minimization) while exploiting at best their resources according to their management and business targets (e.g., consolidating resource usage to reduce costs, performing a fair resource utilization to reduce risks of congestion, maximizing the use of resources to increase revenues at the risk of load peaks). The added value of this work lies in considering (i) a composite yet realistic network and cloud infrastructure to evaluate orchestration strategies extending from access network to the core, and including different types of DCs, (ii) a novel retrial mechanism combined with orchestration strategies, and (iii) twofold latency components, i.e., network and processing latency, that are counter-behaving in a composite cloud and network infrastructure.

The paper is organized as follows. Section II gives an overview of the related works in the literature and highlights the position and the novelties of the proposed research. Section III presents the reference scenario and the resource orchestration framework with dynamic VNF placement decisions. In addition, the challenges of the orchestration process are discussed for the considered scenario. In Section IV, we present the orchestration problem formulation. We also describe the orchestration strategies for the joint selection of computational and network resources and the retrial procedure. Section V presents the simulation design and results. Finally, Section VI provides the concluding remarks for this paper.

II. RELATED WORK

The problem of embedding virtual networks in a substrate network represents the main resource allocation challenge in network virtualization [10]. This is usually referred to as the Virtual Network Embedding (VNE) problem [11]. A wide range of algorithms have been proposed in literature to optimally map virtual networks into substrate infrastructures while assigning resources to provide customized end-to-end guaranteed services to end users [12]. This optimality can be accomplished from different perspectives. More specifically, [13] tackles the embedding problem by providing a solution that simultaneously aims at saving cost, energy and avoiding network congestion. [14] proposes a modified genetic algorithm as a near optimal solution to increase the acceptance ratio and the average revenue of Infrastructures providers (InPs). [15] addresses the survivability of the VNE while improving the network load balance and revenue.

Finally, [16]–[18] consider the energy-efficiency objectives while deciding the optimal VNE.

The VNF allocation problem in DC/cloud environments has also been addressed in several works while orchestrating network and computational resources to set-up network slices (i.e., VNF placement problem). VNF placement and scheduling in public cloud networks is addressed in [19] where a cost-efficient algorithm is proposed to achieve a trade-off between computing resource consumption and latency performance. In [20], authors consider a multi-DC environment where they aim at maximizing the number of accepted VNF placement requests while differentiating them in priority levels and focusing on business related objectives. They also try to satisfy users requirements in terms of latency and minimum guaranteed bandwidth, although only considering link propagation latency. Another related work [21] proposes an optimal algorithm that provides enhanced networking services to Internet of Things (IoT) applications in the Edge and Cloud infrastructures while considering the specificities of each infrastructure (i.e., low latency with limited resources in the Edge against high delay and large scale capabilities in the cloud). Mostly, all these works take into account latency as a global metric to be optimized and do not analyze the effect of its different components (i.e., processing latency, propagation latency) on the placement process, as this work does also thanks to a more composite and realistic reference infrastructure scenario.

Several works address the VNF allocation problem in a single DC [22]–[24]. In particular, in [22], authors present a methodology to make effective VNFs placement decisions by (i) leveraging flexible CPU allocation to VNFs, (ii) addressing arbitrary VNF graphs, (iii) assuring to have multiple instances of the same VNF, and (iv) taking into account the available link capacity and link delays. Although in principle the approach can be adapted to a distributed multi-DC scenario, the methodology is conceived to be applied to multiple virtualized hosts within a cloud DC. Moreover, [25] proposes an optimization framework which allows fine-grained resource allocation for slices both in terms of network bandwidth and cloud processing. In [26], authors present a novel framework to manage the orchestration of end-to-end-slices that jointly perform admission control and resource reservation across different domains while exploiting the concept of slice overbooking to maximize the profit of infrastructure providers.

The ETSI MANagement and Orchestration (MANO) architecture provides a reference framework to regulate and automate operations for network slices in the form of Network Service (NS) deployments according to directives specified in the Network Service Descriptor (NSD). Recently, the MANO framework has also been extended to support the management of multi-site (i.e., multi-DC) Network Services [27] and to consider the case of service deployments in distributed DCs. In such a case, connectivity among virtual computing and storage resources, i.e., VNFs, is necessary across wide area transport networks, e.g., optical or

access networks. The framework also foresees the capability to perform dynamic and optimized VNF placement and resource allocation [28]. Following such guidelines, a number of orchestration platforms have been developed, such as OpenSource MANO (OSM) [29], Cloudify [30], Open Network Automation Platform (ONAP) [31]. However, the full set of features foreseen for the whole MANO framework is still far to be accomplished due to its high complexity and huge number of features. For this reason, several European Projects have also tackled the resource orchestration issue to provide orchestrator prototypes with more advanced features. More specifically, in [32] the placement of the VNFs is performed according to the real-time profile of the deployed application. If the profile is not a-priori known, the VNF is placed near the edge as a de-facto decision. In [33], resource and service orchestration are tackled to support diverse service requirements of various vertical industries where also orchestration strategies have been embedded and tested [34]. Although a significant progress has been done, a full-fledged ETSI-compliant MANO platform is still far to be demonstrated, including the dynamic orchestration operations.

The above research works address VNFs placement and, more generally, effective resource allocation and orchestration in different scenarios. However, as far as our knowledge, all these approaches reveal the lack of extensive investigations on dynamic orchestration including end-to-end latency-related considerations. Latency is instead an important requirement when deploying network services to properly serve emerging applications as stated in [2].

For a comprehensive analysis of the impact of latency on the embedding decisions, both network and processing latency contributions should be considered. In this regard, in [35], the network and processing latency requirements are checked while allocating VNFs at the edge of a fixed mobile convergent network. However, the placement optimization is carried out in a static way from the service provider's perspective, i.e., in terms of resulting consolidation of VNFs, without assessing the slice request admittance and their latency performance. In [36], authors showcase an optimal provisioning of service chain requests with end-to-end latency considerations in physical hardware appliances distributed in a metropolitan network. Both VNFs' processing time and the propagation time over the transport network are considered for the placement decision. The demo is however limited to 2 datacenters with infinite capacities. In [37], authors propose an algorithm for dynamic VNFs provisioning and investigate how to perform traffic grooming in order to address the latency requirements in metro area networks. The algorithm tries to minimize the number of nodes hosting the VNFs while considering their computational capacity and the end-to-end latency as a sum of propagation delays and fixed processing delays. [38] studies the VNF placement problem on edge and public clouds while satisfying user's delay requirements and minimizing the maximum links load. As the problem is NP-hard, an approximation algorithm is proposed leveraging a randomized rounding technique and assuming that the paths

between each node pair are pre-calculated beforehand and given.

Regarding the optical network scenario, several works addressed the allocations of VNFs along with optical network paths. In [39], authors study the deployment of VNF service chains over flexible-grid elastic optical networks tackling several aspects of the problem (traffic grooming, spectrum continuity and contiguity, VNF placement) with the objective of reducing network energy consumption. In [40], authors investigate the VNFs selection problem to achieve joint load balancing of IT and spectrum resources trying to achieve an efficient provisioning of the service requests. Both works do not take into account the DCs peculiarities in terms of location and processing capabilities as this work does. On the contrary, the impact of VNFs location is studied in [41] where authors provide a genetic algorithm to solve the problem of dynamic VNF placement in a metro optical network. The solution takes into consideration the VNF location by looking for available resources first at the MEC nodes then at the central office and finally at the rest of nodes of the network. The aim is to reduce the service blocking ratio while optimizing the use of both the computational and optical resources. As a solution to further minimize the blocking, the authors consider a collaborative method between MEC resources, while in this work we propose a retrial mechanism. Finally, the techno-economic impact of VNFs placement on the optical metro network cost is studied in [42]. Authors perform a comprehensive analysis of several VNF placement strategies considering service chains requirements such as bandwidth and end-to-end latency. They conclude that an efficient strategy can reduce the operational cost in terms of delay and Quality of Experience.

As highlighted in [43], the benefits brought by retrial mechanisms have been demonstrated in several fields, e.g., call centers [44], cognitive networks [45] and cellular networks [46]. In those retrial mechanisms, unsatisfied customers are temporarily stored in a queue and then superimposed on the normal stream of new arrivals. Each blocked customer then generates a source of repeated requests to try to fulfill its service independently of the rest of the other requests which might improve the number of satisfied customers. Motivated by the benefits brought by retrial mechanisms, we applied them to slice service demands to increase requests acceptance rate and, hence, telco service provider's revenues.

To summarize, and, to the best of our knowledge, this work advances the state of the art by considering a sophisticated scenario, characterized by a composite yet realistic reference infrastructure (i.e., wide infrastructure from access to core network and different DCs capabilities based on their network location), different set of parameters (i.e., twofold latency components, specificities of path selection in flexible optical networks) and novel adopted mechanisms in orchestration (i.e., retrial). This paper extends the preliminary work presented by authors in [47], [48] with (i) the proposal of novel orchestration strategies accounting for the end-to-end latency,

(ii) a more general reference scenario not only comprising virtualization of mobile network functions, (iii) a more comprehensive evaluation of the orchestration strategies using different performance indicators, and (iv) a comparison with opaque networks (i.e., with optical-electro-optical conversion in each node). This work also extends the analysis carried out by authors in [49] where a subset of the orchestration strategies presented in this paper have been assessed in a laboratory set-up. In particular, we supplement that analysis by extending the set of orchestration strategies considering also resource management targets (i.e., load balancing) and using simulations in a more composite network and DC scenario and with a larger scale, thus coming to more general conclusions.

III. REFERENCE SCENARIO AND CHALLENGES

This section introduces the reference scenario we considered and the building blocks we elaborated to evaluate the resource orchestration strategies. Moreover, the challenges of the orchestration problem are discussed.

A. REFERENCE SCENARIO

The reference scenario is shown on the bottom of Fig. 1 and consists in a typical set-up of a 5G network infrastructure. More specifically, it considers a wide set-up of geographically-distributed clouds (i.e., DCs) interconnected through edge, metro and core network links.

DCs span from large cloud DCs to resource-constrained DCs located at the network edge to address network latency requirements of delay-sensitive 5G applications. According to [50], we can contemplate the scenario in which, in relation to their network locations, the DCs feature not only distinct sizes but also different processing capacities and hence different latencies while processing data. More precisely, DCs can be classified in:

- *Edge DCs*: They are pervasively deployed at the network edge and co-located with telco PoPs in the access segment to exploit the proximity to users and offer minimized network latency performance. They have constrained processing capacity because they are small-sized DCs and typically feature higher processing latency because they deploy low-power computing systems tailored to support only users connected to the access network;
- *Metro DCs*: They are located in the metro or regional network segment and feature average processing capacity and processing latency performance;
- *Core DCs*: They are large-sized cloud DCs, deployed typically in few remote locations beyond core network segments and thus reachable at the expense of increased network latency performance. They have substantial processing capacity because of their large scale and typically feature lower processing latency because they deploy high-powerful computing systems tailored to support a huge amount of users while taking advantage from economy of scale.

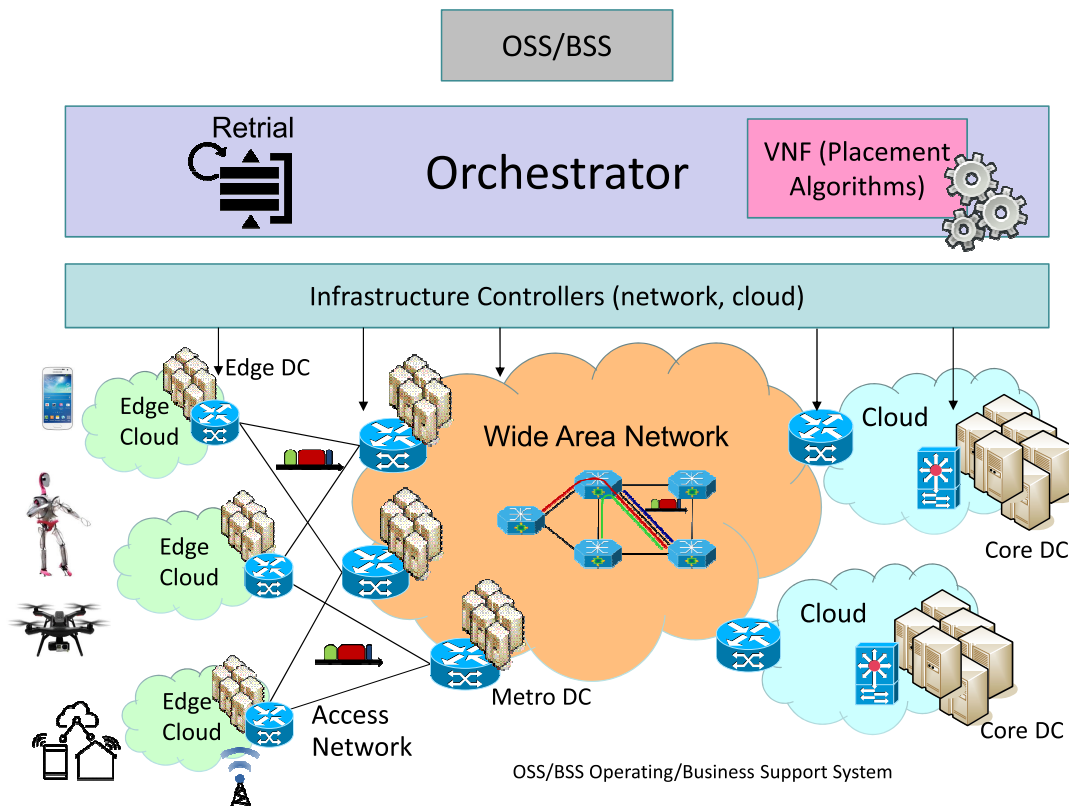


FIGURE 1. Reference network scenario and building blocks of dynamic resource orchestration.

As a case study, we considered DCs interconnected through a convergent all-optical wide area network, spanning from the access/metro segments to the core network, thereby constituting a telco cloud infrastructure [51], [52]. The optical network connects cloud DCs and arrives up to the access, feeding the mobile networks (i.e., 5G radio access) and the network edge (i.e., telco *edge DCs* acting as PoPs). The all-optical wide area network supports flexible grid technology to enable the elastic and dynamic allocation of optical network bandwidth for each virtual link connecting VNFs.

“Flexi-grid” technology or Elastic Optical Network (EON) has been proposed as an interesting deployment option of traditional optical networks with rigid frequency grid to effectively address dynamic and bandwidth intensive traffic demands of Internet-based applications [53]. Optical networks supporting a flexi-grid allow to accommodate data traffic through dynamic allocations of “just-enough” optical spectrum leveraging the configurability of sliceable bandwidth variable transponders (SBVTs) [54]. More specifically, the available optical spectrum can be divided into a set of frequency slots (FSs) of a fixed (finer) spectral width, e.g. 25GHz, 12.5GHz or even 6.25GHz, in comparison to the current ITU-T Dense Wavelength Division Multiplexing (DWDM) rigid frequency grid (e.g. 50GHz) [55]. The number of contiguous FSs then depends on the requested bitrate, the modulation technique and the slot width. This allows

bandwidth to be dynamically and elastically reserved for each connection with a finer granularity by adding or releasing FSs to accommodate changing requirements of traffic demands [56]. Moreover, flexi-grid optical networks allow for optical grooming techniques to be applied to data traffic thereby offloading the electronic processing burden to the optical layer [57].

Flexi-grid optical networks still allow the adoption of Optical-Electrical-Optical (OEO) conversion at every node in the network, which in some scenarios further improves the blocking and failures detection. More specifically, flexible grid optical networks offer the possibility to setup the connections in two ways: transparent and opaque. For the transparent scenario, the signal propagates all-optically from source to destination throughout the network. On the contrary, in the opaque scenario each optical channel is converted to the electrical domain and then back to the optical at every traversed node in the network. Depending on the scenario in which they are considered, both solutions present benefits (e.g., power savings by eliminating unnecessary opto-electronic conversions in transparent networks versus supporting fault detection, isolation and performance monitoring in opaque networks) [58].

In combination with SDN, the flexi-grid optical networks promise to effectively address the agility and flexibility demanded by NFV-enabled multi-DC scenarios. Indeed, they

assure the elastic and dynamic allocation of network bandwidth for each virtual link connecting VNFs while enabling efficient bulk data transfers across DCs and lower latency performance [59]. Hence, orchestration platforms can be effectively put in operation by exploiting the programmability offered by SDN controllers on top of flexi-grid optical networks in distributed DC environments [53].

B. ORCHESTRATION BUILDING BLOCKS AND CHALLENGES

The *network softwarization* forces a consolidation of the optical and mobile/access network and a convergence of the telecommunication network with the computational resources in the DCs, leading to the challenging task of selecting and coordinating the allocations of both (virtual) network and computational resources to serve slice requests across distributed DCs while meeting requirements specified for network slices. This is the task carried out by the orchestration system (i.e., Orchestrator) shown on top of Fig. 1.

The Orchestrator elaborates slice requests from the Operating/Business Support System (OSS/BSS). The OSS/BSS is in charge of service delivery and business operations to fulfill service requests from customers (e.g., vertical industry). Upon the arrival of a slice set-up request, the Orchestrator (i) selects the appropriate virtual resources across DCs and interconnection network to allocate and deploy VNFs as well as virtual links composing the network slice (i.e., VNF placement decisions) while addressing computational and network requirements (i.e., bandwidth demand at virtual links, number of CPU cores to run the VNFs); (ii) sets-up the virtual machines hosting VNFs in the selected DCs and the network paths (both legacy and SDN-enabled) in the interconnection network while leveraging cloud and network controllers, and (iii) assures the proper operation of VNFs and network slicing during their lifecycle through monitoring.

The VNF placement decisions are made dynamically (i.e., on per-request basis) and different orchestration strategies can be adopted depending on the resource and service management targets of telco service providers. The orchestration strategies run on a system-level view including the current status and load of the underlying resource infrastructure (e.g., infrastructure topology, available resources, offered latency performance) to take appropriate VNF allocation decisions.

In the view of evolved interworking with the OSS/BSS (and ultimately with vertical industry) in highly dynamic 5G scenarios, retrial mechanisms are considered at the Orchestrator in case of not successful VNF deployments due to the lack of virtual resources in order to allow for multiple slice set-up attempts and increase the rate of deployed slices although at the cost of increased deployment time. The strategies and the retrial mechanism are detailed in Section IV.

A number of challenges need to be faced while doing orchestration of cloud and network resources on top of a 5G infrastructure. Toward maximizing the acceptance rate,

the challenge is to properly exploit the available resources in the infrastructure to serve as many slice requests as possible while addressing specified requirements. For this reason, the orchestration strategies evaluate load information at both the DCs and the network links interconnecting them. The load information is especially critical in relation to *edge DCs* featuring highly constrained computing resources and at the same time being strategic for latency-sensitive applications due to their proximity to users. Hence, it is advisable that load information are used as a criteria to select DCs or network paths also in consideration of target management goals of telco operators. For instance, the load can be considered in the orchestration decisions to minimize the risk of congestion (and hence minimize the risk of latency performance degradation) while pursuing a fair resource utilization (e.g., load balancing, proportional allocation).

On the other hand, the load might not be a primary criteria in the orchestration decisions if the operator management goal is to consolidate resource usage (and hence reduce operational costs). In such a case, the latency minimization can be achieved through DC location-based considerations (e.g., selection of the closest DC). However, disregarding load information in the orchestration decisions may come with the risk of DC or network link overburdening (and hence the risk of latency performance degradation). In any case, a check that the DC or link nominal capacity value is not exceeded is advisable before admitting the requests, thus leading to the possibility that requests are blocked. For this reason, the retrial mechanism can be beneficial to reduce the rate of slice request blocking. However, the retrial comes with a higher slice acceptance and deployment time. Hence, the challenge while doing retrial is to mitigate this additional time through much more effective usage of resources in order to resort on retrials as less as possible.

On top of that, the challenge to address end-to-end latency minimization in the considered scenario is also handling twofold latency components and their inter-play depending on considered DC network location. More specifically, for a comprehensive orchestration decision in the presented scenario, the following latency components should be considered, (i) the network latency experienced by data traversing the DCs interconnection network links, and (ii) the processing latency experienced by data at the DCs servers. Such latency components are counter-behaving. Indeed, DCs feature opposite network and processing latency performance based on their location, whether in the access, metro, or core network segments. More specifically, small DCs at the edge network offer higher processing latency due to their low-power computing systems while being reachable with lower network latency. On the contrary, large DCs beyond the core network offer lower processing latency thanks to their high performance systems but impose higher network latency due to their distant location.

Additional challenges raise up in doing orchestration over flexi-grid optical interconnections. A challenge to efficiently use EON resources for network path set-up is related to

the physical layer impairments while applying the routing procedure to interconnect any couple of DCs. More specifically, some metrics such as the transmission distance and the number of traversed nodes should be considered while determining the candidate paths. Some approaches such as the use of k -shortest link-disjoint paths have been proposed, mainly to relax the blocking rate of lightpath requests. More specifically, k alternative candidate routes can be calculated in order to be considered if the shortest path for a given source-destination pair is not available. This approach, along with an efficient orchestration strategy can, in some scenarios, significantly improve the performance of EONs. In fact, having more paths available can have a significant impact on the overall acceptance rate and especially on the inter-DC blocking component, thus alleviating the blocking and improving the performance.

Secondly, flexi-grid networks require high OEO conversion capabilities to be able to perform the switching of channels with variable bandwidth characteristics at a fine granularity [60]. However, the adaptation to the changing traffic and network conditions is limited by the reconfigurations allowed by the optical switches and frequency-tunable lasers. In this context, the key to alleviate the stress on the optical nodes is then to adopt orchestration strategies, which can efficiently assign the available resources while minimizing the OEO conversion cost (e.g., through the adoption of transparent optical networks).

Finally, it is worth mentioning that also bandwidth fragmentation represents a serious issue in the set-up of network paths across elastic optical networks. In fact, the constraint of spectrum continuity (i.e., the same FSs must be used in all links of the path) and contiguity (i.e., the allocated FSs must be contiguous in the spectrum) is hard to respect with the dynamic provisioning and releasing of the optical paths in the network, hereafter called “lightpaths”. Moreover, in such a context, non-contiguous and non-aligned free FSs are more likely to be created thus increasing the impossibility to satisfy new lightpath requests and hence increasing the blocking rate in the network [61]. Such fragmentation issue can be overcome in two ways: either by using wavelength converters that however bring an additional deployment cost to the service providers, or by adopting proper wavelength/spectrum assignment (WA/SA) strategies that enhance the accommodated resources demands while limiting their fragmentation impact [62]. In this paper, we assume that the number of frequency slots is fixed for all the lightpaths and we choose not to focus on this fragmentation issue leaving it for further insights from network designers in optical networks.

IV. LATENCY-AWARE RESOURCE ORCHESTRATION WITH RETRIAL

This section introduces the latency-aware orchestration problem on top of a distributed telco cloud infrastructure for 5G network slicing. By “latency awareness”, we mean that the orchestration strategies take into consideration the latency performance offered by the infrastructure (and more

precisely, two latency components that are propagation and processing latencies) during the decision process to select the most suitable data center that minimizes the end-to-end latency. In the following subsections, we firstly present the formal problem to address when dynamically processing incoming slice requests along with the system model. Then, a description is provided of the three different orchestration strategies we propose in combination with a retrial mechanism.

A. ORCHESTRATION PROBLEM FORMULATION

We modeled the distributed telco cloud infrastructure as a weighted directed graph $G(\mathcal{V}, \mathcal{E})$, where \mathcal{V} represents the set of network nodes and \mathcal{E} represents the network links. Each link in \mathcal{E} is defined as a tuple $\{u, v, dist_{uv}, availBw\}$ where $dist_{uv}$ represents the distance from node u to node v and $availBw$ represents the available bandwidth on the link expressed in terms of FSs.

In a subset of nodes $\mathcal{V}^{DC} \in \mathcal{V}$, a data center is co-located with the network node. Every data center dc can be described as a tuple $\{ID, t, cr_{dc}, L_{dc}\}$ where ID is the data center identifier, t indicates the data center type (edge, metro or core), cr_{dc} represents the computational resources that can be allocated for running the virtualized VNFs and L_{dc} represents the processing latency of the data center dc .¹

Upon a slice request, the Orchestrator coordinately selects both the DCs where to allocate the computational resources needed by the VNFs (e.g., required processing or storage capacity) and the network resources needed to interconnect them according to the bandwidth demand (e.g., required FSs in the flexi-grid WAN).²

We assume a forwarding graph composed of a source VNF connected by a virtual link to a destination VNF. Accordingly, the Orchestrator selects a source DC and then a network link and the destination DC, while addressing the requirements of the request. In such a way, every slice request has two vertices (i.e., two VNFs to allocate in as many source and destination DCs) and one edge (i.e., one VNF interconnection link to allocate across the flexi-grid optical network). The reason for such an assumption lays in our main focus on evaluating the effectiveness of the orchestration strategies in fulfilling slice service requirements while minimizing the end-to-end latency and while considering constraints and inter-dependencies given by the composite yet realistic reference scenario described in Section. III-A. More specifically, the focus is on the impact of the combined selection of (i) the virtual link as result of a complex routing process in the underlying flexible optical network while offering different network latency performance, (ii) the DCs where to

¹We consider the processing capacity, to which we refer as computational resource, as the only relevant parameter that allows us to differentiate the performance among edge, metro and core DCs. We investigated the combined requests for computational, storage and memory resources in [49], [63].

²Without loss of generality, it is assumed that the mapping between the required transmission rate and the number of FSs is performed offline and thus the requests are characterized directly in terms of FSs.

deploy the VNFs resulting in different and counter-behaving processing and network latency performance as explained in Section. III-B. In addition, the focus is also on the coupling of the above orchestration key elements with the retrial mechanism that is highly affected by the amount of resource utilization. In such an assumption, the VNF source and VNF destination can represent a cluster of VNFs arbitrary interconnected among each other although still placed in the same DC. The generalization to the case of arbitrary slice requests is left for future work.

Finally, regarding the source DC, we select it under the assumption of location constraints being specified in advance, e.g., the access connection of the users to be served and/or the user location [49].

Accordingly, let R^i be the slice request which can be viewed as a tuple $\{i, s, \xi, \beta\}$ where i represents the request identifier, $s \in \mathcal{V}^{DC}$ represents the source DC, ξ the amount of requested computational capacity, and β the required bandwidth expressed in terms of number of FSs. The number of available FSs can be expressed as follows:

$$N_{FS} = BW_{fiber\ link} / FS_{spectrum\ width} \quad (1)$$

Following the same logic, each request R^i demands a given amount of bandwidth, which can be translated into a number of FSs as expressed in the following equation:

$$N_{FS} = \left\lceil \frac{Req.BW\ (GHz) + Guard\ band\ (GHz)}{FS_{spectrum\ width}} \right\rceil \quad (2)$$

The requested number of FSs for each request is equal to the division between the requested bandwidth and the spectral width of a single FS.³

A request is blocked if the selected destination DC has no available resources, the source DC has unavailable resources or the network resources along all the paths from the source to the selected destination DC are unavailable.

Let B_{dc} be the probability that data center dc does not have any available resource for supporting the request from s . Let \mathcal{P}_{sd} be the set of paths computed from s to d for supporting R^i and B_{sd}^p be the probability that there are no bandwidth available along the paths from the source node s to the selected destination node d . The blocking probability for a request from node s is thus equal to the probability that the DC source does not have any available resources plus the probability that there are no available resources along any network path between the DC source and DC destination:

$$B = B_s + (1 - B_s) \prod_{p \in \mathcal{P}_{sd}} \left[B_{sd}^p + (1 - B_{sd}^p) \cdot B_d \right] \quad (3)$$

where B_s expresses the probability that the DC in s does not have available capacity and if it has it, either a path p with available bandwidth cannot be found (i.e., B_{sd}^p) or the DC in d does not have available capacity (i.e., B_d). In fact, in this paper, we consider two possible blocking components

³Some technologies such as Bandwidth Variable Wavelength Cross Connects (BVWXC) require guard bands between signals to perform the switching adequately, which should be added to the requested bandwidth [64].

which are, namely, the *inter-dc* blocking referring to path unavailability between a giving source-destination pair of data centers and the *intra-dc* blocking referring to resources unavailability on the data center source and/or destination.

The end-to-end latency experienced by a VNF allocated at d and connected to node s (L_{sd}) is computed as follows:

$$L_{sd} = \sum_{\ell \in p_{sd}} L_{\ell} + L_d \quad (4)$$

where L_{ℓ} is the *propagation latency* on link ℓ from s to d and L_d^{DC} is the *processing latency* of the DC at d . For each data center dc in the network, the processing latency is expressed as follows:

$$L_{dc} = Data / [Processing_Capacity]^{dc} \quad (5)$$

where *Data* refers to the amount of data to be processed and *Processing Capacity* refers to how many operations can be carried out by the data center dc in a given amount of time.

Regarding the use of retrial mechanisms to increase the acceptance rate of slice set-up requests, the orchestration process can be modeled as a system where blocked customers temporarily leave the service and then try again after some random time. Those blocked customers are ranked in a waiting queue before retrying to be served again. The correspondent retrial queuing model is presented in the following. We consider c servers (i.e., c refers to the number of DCs that might satisfy the request) and no waiting queues before the server. Requests arrive at the system according to a Poisson process with rate λ and the service time for every request has an exponentially distributed duration with mean $1/\mu$. An arriving request is immediately served if its requirements can be satisfied by one of the data centers in the network. Otherwise, the request is put in the waiting queue and retried again after a retrial interval.

In this M/M/c queue model, let $\Pi_m(z)$ be the generating function of the number of requests in the retrial queue:

$$\Pi_m(z) = \sum_{j=0}^{\infty} (\pi_{m,j} z^j) \quad (m = 0, 1, \dots, c) \quad (6)$$

where $\pi_{m,j}$ denotes the joint probability that there are m ($m = 0, 1, \dots, c - 1, c$) not available DCs and j blocked requests ($j \in \mathbb{Z}^+$) in the retrial queue [65].

As explained in [43], explicit expressions for $\pi_{m,j}$ and $\Pi_m(z)$ can be found for the cases $c = \{1, 2\}$ while it is challenging to obtain analytical solutions for the cases with ($c \geq 3$).

B. ORCHESTRATION STRATEGIES AND RETRIAL PROCEDURE

The orchestration strategies run under a common algorithmic approach for VNF placement, but featuring different objectives, i.e., different selection criteria.

The pseudo-code of the algorithmic approach is presented in Algorithm 1. The algorithm takes the graph $G(\mathcal{V}, \mathcal{E})$, the request R^i and the orchestration strategy as input.

Algorithm 1 Orchestration Strategies

Data:
 $G = (\mathcal{V}, \mathcal{E})$ // Graph of the network and DC infrastructure
 Select the user location randomly among DCs
 s = User location
 $R^i = (i, s, \xi, \beta)$ // Request
 $\mathcal{Q} = \{\emptyset\}$ //Queue for retrial
 $\mathcal{D} = \{\emptyset\}$ //Set of DCs
 $\Pi = \{\emptyset\}$ //List of candidate paths
 Select one orchestration strategy from {MD, ML, LB}
Begin procedure
if *orchestration strategy is MD* **then**
 \mathcal{D} = Sort the DCs in increasing order of geographical distance.
 Select candidate DC d from \mathcal{D} .
else
 if *orchestration strategy is LB* **then**
 \mathcal{D} = Sort the DCs in increasing order of load.
 Select candidate DC d from \mathcal{D} .
 else
 if *orchestration strategy is ML* **then**
 \mathcal{D} = Sort the DCs in increasing order of the overall
 latency, i.e., propagation plus processing latency in
 DC.
 Select candidate DC d from \mathcal{D} .
 else
 Compute the k -shortest paths from s to d : $\mathcal{P} = \{p_1, p_2, \dots, p_k\}$
 for $path\ p \in \mathcal{P}$ **do**
 find continuous and contiguous FSs f_h on p
 if $\sum_{h=1}^N f_h \geq \beta$ // Enough bandwidth on p
 then
 put p in the list of candidate paths Π
 if $\Pi = \{\emptyset\}$ **then**
 No available candidate paths found to d . Put request in \mathcal{Q} for
 retrial.
 else
 while ($path\ p \in \Pi$) and (R^i is not accepted) **do**
 if ($cr_d \geq \xi$) and ($cr_s \geq \xi$) **then**
 R^i is accepted
 update availability of FSs f_h on p
 $cr_d = cr_d - \xi$ //Update computational resources in d
 $cr_s = cr_s - \xi$ //Update computational resources in s
 if R^i is not accepted **then**
 No available resources. Put request in \mathcal{Q} for retrial.
 End procedure

Before applying the algorithm, the retrial queue \mathcal{Q} , the set of DCs \mathcal{D} selected as candidate for deploying the VNFs and the list of candidate paths between the source and destination DCs are empty. Moreover, we select the user location randomly among the set of DCs and we assign it to the source DC. The algorithm starts by selecting the best DC according to the selected orchestration strategy as explained below. For each candidate DC d , the k -shortest paths from s to d are computed considering links length as a metric [66]. The algorithm goes through the list of these k -shortest paths sorted by distance to check the availability of network resources. More specifically, for each path p , the spectrum contiguity and continuity and non-overlapping constraints are checked and if the spectrum is available to accommodate the request, the path is added to the list of candidate paths. If no available FSs are found on any of the k -shortest paths then the request is put on the retrial queue for another attempt later. Otherwise, if FSs are available for the requested requirement,

the availability of ξ computational resources is checked in the source DC s and in the destination DC d . If both DCs have sufficient resources, the request is accepted and then the statuses of the resources in the network and in the DCs are updated. Otherwise, the request is put on the retrial queue for another attempt later.

Based on the adopted selection targets, we devised three different categories of policies that aim at the minimization of the latency performance while considering the following criteria: (i) DC proximity, (ii) DC end-to-end composite latency capability and (iii) fair resource utilization. Under each category of policies, there might be several algorithms or selection criteria to implement. In this work, we propose one strategy for each category as an example of implementation. More specifically, DC proximity is implemented through the *Minimum Distance strategy (MD)*, which minimizes only the network latency component. The DC end-to-end composite latency capability is implemented through the *Minimum Latency (ML)* strategy, which minimizes the composition of network and processing latency. Finally, the fair resource utilization is implemented through the *Load Balancing (LB)* strategy.⁴ The idea behind LB is that such a strategy could also have an indirect benefit also in terms of latency offered by the DC systems and interconnections while pursuing a balanced resource usage. More specifically, the three orchestration strategies behave as follows:

- *Minimum Distance (MD)*: This is the reference strategy which aims at favouring the closest DCs, thus achieving the lowest propagation latency. It selects the closest DC (in terms of path length) provided that it still has available computational resources and provided there are still available network resources in the WAN to set-up a network path to reach it with the required bandwidth demand. This strategy is used as a baseline for comparison.
- *Load Balancing (LB)*: In LB strategy, the goal is to balance the load among the DCs thereby achieving a fair resource management and utilization. The load is defined as the total amount of computational resources reserved for the requests that are currently served by the DC, over the amount of computational resource of the DC. It selects the DC that has the lowest load (i.e., the highest amount of available DC resources) among those with available computational resources and the network paths in the WAN with most available network resources among the k pre-computed paths.
- *Minimum Latency (ML)*: In ML strategy, the aim is to favor the DC which achieves the lowest end-to-end latency and thus optimizes the delay perceived by the user/application. For each candidate destination DC, it computes the sum of the propagation latency and the processing latency. Then, it selects the destination DC that minimizes the sum of the two latency components.

⁴The proportional allocation could be another valid option for the fair resource utilization category, resulting in a load distribution outcome among available resources.

Algorithm 2 Retrial Procedure

Data:
 \mathcal{Q} // Queue of non satisfied requests
 st_i //step of the request R^i

Begin procedure
for $R^i \in \mathcal{Q}$ **do**
 $st_i = st_i + 1$ //update timer for every request in the queue
 Apply Algorithm 1 for R^i
 if R^i is accepted **then**
 Remove R^i from \mathcal{Q}
 else
 if $st_i \geq WT$ **then**
 Remove request R^i from \mathcal{Q}
 R^i is rejected
 end if
 end if
end for

End procedure

In the following, we describe the *retrial mechanism*. When a request cannot be promptly satisfied, it is further delayed, by storing it in a FIFO queue of waiting requests. Each waiting request has a maximum Waiting Timeout (*WT*). Upon expiration of *WT*, the request is dropped by removing it from the FIFO queue and is considered rejected. When resources are released (e.g., termination of other requests), a retrial procedure is performed on the requests in the queue. This retrial can be repeated until either the request is satisfied or its associated *WT* expires.

Algorithm 2 describes the retrial procedure which takes as input the queue of non-satisfied requests and the current status of the timer for every request in the queue. The retrial procedure starts by updating the timer of all the requests in \mathcal{Q} , then for every R^i tries to apply Algorithm 1 to check if there are any available resources to accommodate some or all the requests of \mathcal{Q} . If any request R^i is accepted, it is then removed from \mathcal{Q} . Otherwise, the current timer st_i of every request R^i in \mathcal{Q} is checked and compared to *WT*. If st_i is lower than *WT*, R^i is kept in the queue for a check at the next round, otherwise it is removed and the request is permanently rejected.

V. PERFORMANCE EVALUATION

In this section the orchestration process is evaluated. The simulation environment setup is firstly presented, then subsequent subsections illustrate the performance of the orchestration strategies (Section V-A), the additional benefits provided by the adoption of the retrial mechanism (Section V-B), and the impact of design choices in terms of network routing (Section V-C) and Flexi-grid Network transparency (Section V-D) on the efficiency of the orchestration process.

For the evaluations, we used a custom-made event-driven simulator implemented in Java and running the proposed strategies on top of an infrastructure scenario as described in Section. III-A. The source code of the simulator can be found in [67]. The distributed DC infrastructure is reproduced that consists of six *edge DCs*, three *metro DCs* and three *core DCs* connected according to the network topology shown in Fig. 2.

The considered topology combines both the core and the edge network. The core network interconnections are inspired by a real telco network topology in Germany [68] with 17 nodes and 26 links. The edge network interconnections are inspired by a typical tree-like topology with redundant links among small DCs and medium DCs. Links lengths are reported in km above the connection lines for both edge and core networks.

Moreover, we consider that each *edge DC* has a specific processing capacity fixed to 7.5 GB/s while the processing capacities of the *metro* and *core DCs* are calculated as 2 times and 4 times the value of the edge one, respectively [50].

To obtain the processing latency values, we apply Equation 5. For this purpose, we need to fix the amount of data the application is supposed to process. This value depends on the applications using the deployed network slices. The values might go from few kilobits (e.g., web service use case) to hundreds of megabits when bulk data traffic is considered (e.g., Content Delivery Networks) [69]. In this paper, we consider an average value equal 250Mb for each request, which brings to the following values for the processing latency for each type of DC: 4.16ms for small DCs, 2.08ms for medium DCs and 1.04ms for the largest one.

As for DC network interconnections, an optical network with a flexible grid is considered with fiber link lengths expressed in kilometers as reported in Fig. 2. Each fiber link supports 320 frequency slices of width 12.5GHz. The transmission rate is 100Gbps, requiring 4 frequency slices for each lightpath. Full wavelength conversion capabilities are assumed to be available at each node.

Requests are generated randomly according to a Poisson process. Each request has an exponentially distributed duration. The source DC of each request is uniformly distributed. For symmetry, two types of requests are considered: (i) the source DC is an *edge DC* and thus, the destination DC should be a *metro/core DC*; (ii) the source DC is a *metro/core DC*, whereas the destination DC should be an *edge DC*. Unless otherwise stated, the waiting timeout *WT* for the retrial mechanism is equal to 50s, while routing is performed by considering the first 4 shortest paths (i.e., $k = 4$).

We evaluate the following set of metrics: (i) the *blocking probability* calculated as the percentage of accepted requests over the total number of received requests; (ii) the *processing latency* expressed as the average time necessary for a data center to process all the requested data; (iii) the *queue time* calculated as the average time spent by a rejected request in the retrial queue; and (iv) the *number of retrials* expressed as the number of times the Orchestrator resorts to the retrial mechanism to try to fulfill a request.

Results are displayed as a function of the network load expressed in Erlang (*E*). Each graph is the result of the average of 10 simulations executed with the same parameters. In every iteration, 500000 requests are sent. Results are collected with a confidence interval of 5% at a confidence level of 95%. All the simulations parameters and their corresponding values can be found in Table. 1.

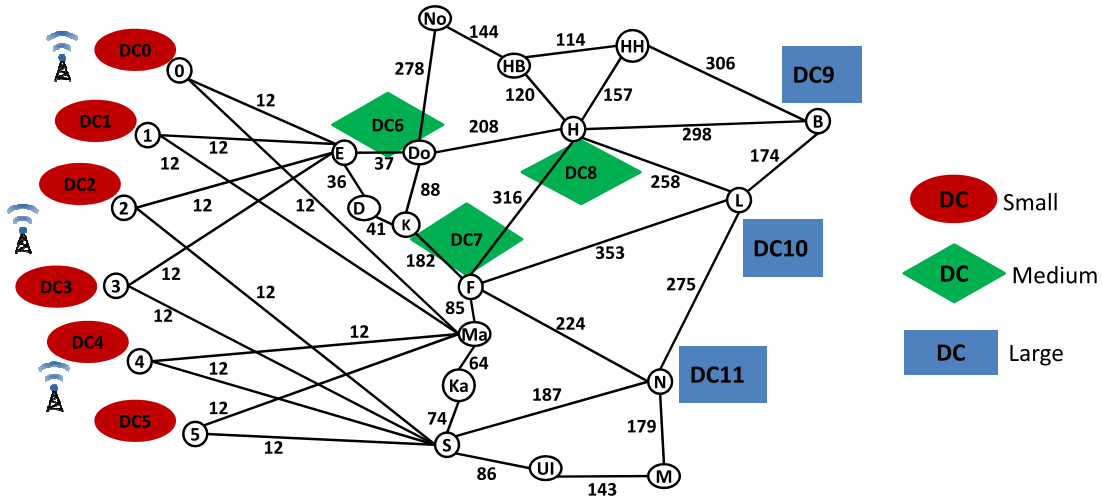


FIGURE 2. Distributed DC infrastructure with flexi-grid network.

TABLE 1. Simulation parameters.

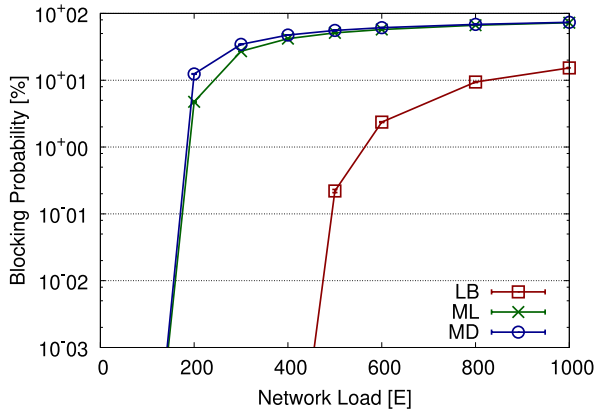
Category	Parameters	Value
Topology	Number of DCs	12
	Number of FSs per fiber link	320
	FS width	12.5GHz
	Transmission rate	100Gbps
Resources demand	Number of FSs per request	4
	Data to process	250Mb
Edge Resource Capacities	DC processing capacity	7.5GB/s
	Processing latency	4.16ms
Metro Resource Capacities	DC processing capacity	15GB/s
	Processing latency	2.08ms
Core Resource Capacities	DC processing capacity	30GB/s
	Processing latency	1.04ms
Simulations	Waiting timeout	50s
	Number of shortest paths	4
	Number of requests per simulation	500000

A. PERFORMANCE OF ORCHESTRATION STRATEGIES

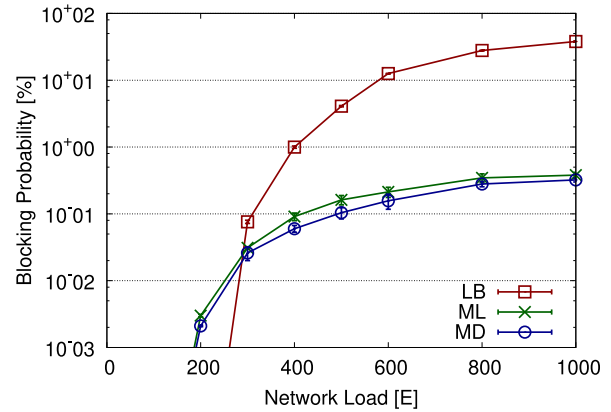
Fig. 4 compares the blocking probability of the three orchestration strategies: MD, ML and LB when the value of k is set to 4. MD selects always the closest DC which increases its rejection rate since it is more likely to not find available resources towards/in the selected DC due to the high utilization of only few closer DCs (e.g., typically the metro DCs in the uplink direction) and thus of the adjacent network links. ML strategy slightly improves the performance of MD at low loads thanks to the fact that distant DCs (e.g., core DCs in the uplink direction) can be preferred for their faster processing time, enabling a slight redistribution of the load among DCs. LB strategy achieves the lowest blocking probability as it balances the load among DCs and thus ultimately balances also the load on the network links.

Fig. 3 quantifies the two components of the blocking probability, i.e., the *intra-DC blocking* due to the lack of DC resources in Fig. 3(a) and the *inter-DC blocking* due to the lack of network resources in Fig. 3(b). Results show that for MD strategy, the intra-DC blocking component is higher since the requests are directed to the closest DC until its resources are exhausted. The same behavior is experienced also in the ML strategy which directs the requests to the DC presenting the lowest overall latency. Since the latency depends on static capabilities, i.e., the distance towards the DC and the DC processing capacity, the results are very similar to those of the MD strategy. On the contrary, in the LB case, the inter-DC component is higher than the intra-DC one, because the LB strategy distributes the load on all DCs which results in a better utilization and a lower consumption of the intra-DC resources. However, spreading the load among different destinations causes also a higher and more fragmented utilization of network resources, leading thus to a significant inter-blocking component in MD and ML.

Fig. 5 plots the average processing latency for the three orchestration strategies as a function of the network load. Since the processing latency is strictly related to the power of the computing systems deployed within the DCs, in this graph we show the impact of the proposed strategies by choosing the DCs taking into account their different capabilities. Results show that MD suffers the highest processing delay since it selects the closest DCs (i.e., metro DCs in the uplink direction) which typically deploy low-power computing systems and thus feature a high processing latency. On the contrary, the LB strategy improves the processing latency of about 25% with respect to MD. In fact, it selects the destination DCs according to their current load. Since the core DCs deploy high-powerful computing systems, they are more likely to be less loaded and support a larger amount of users and thus be selected with respect to metro and edge DCs. Such high-powerful capabilities feature a low processing latency which



(a) Intra-DC Blocking Component



(b) Inter-DC Blocking Component

FIGURE 3. Blocking probability components vs. load.

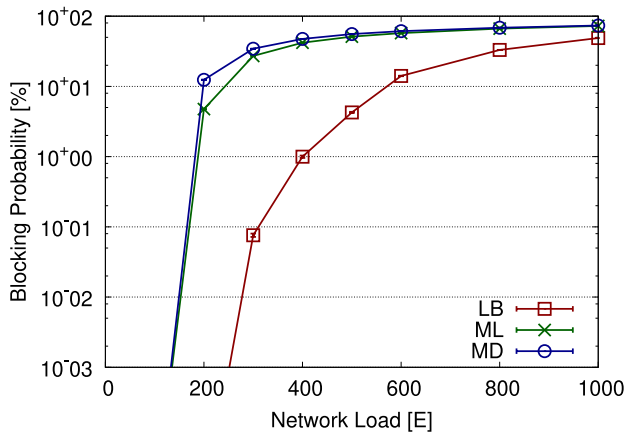


FIGURE 4. Blocking probability vs. load for $k = 4$.

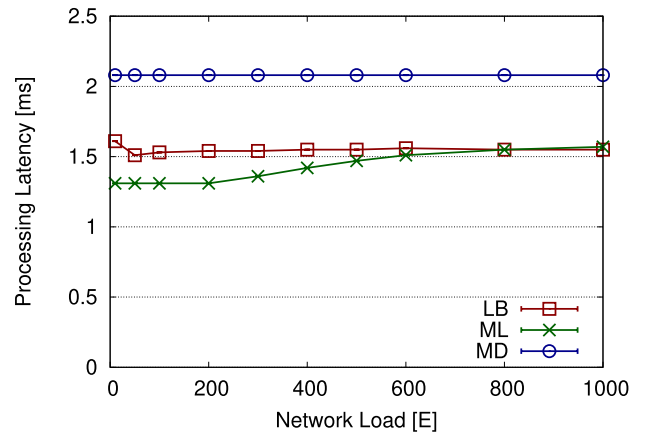


FIGURE 5. Processing latency vs. load.

explains the better performance of the LB strategy. Finally, the ML strategy introduces the lowest processing delay. Such behavior is expected since ML aims at selecting the DC that minimizes the overall latency (i.e., both propagation and processing components). Since the propagation delay is negligible with respect to the processing one, ML always selects the DC that offers the lowest processing latency. It is worth to notice that at high loads, *core DCs* become congested (i.e., unavailable) and ML starts to select *edge/metro DCs* thus leading to an increase in the processing delay up to achieve the level of LB.

B. PERFORMANCE OF ORCHESTRATION WITH RETRIAL MECHANISM

The retrial mechanism is adopted to try to increase the service providers revenues by not directly rejecting slices deployments in case of lack of resources. Hence, such a mechanism does not impact the processing latency while it has a significant impact on the requests acceptance rate. In fact, when it is not possible to deploy the slice immediately, the request

is kept in a queue and retrials for its fulfillment are achieved within a given interval of time named as the maximum Waiting Timeout (*WT*). When *WT* expires, if the virtual resources are still unavailable, the request is blocked.

Fig. 6 plots the blocking probability as a function of the network load for the three orchestration strategies, with and without retrial mechanism. By adopting the retrial mechanism, the blocking rate, as expected, decreases for all the strategies. Indeed, trying to fulfill a blocked request at a later time can be beneficial since the request might be accepted thanks to the release of the needed resources in the meanwhile. The improvement is more marked in LB strategy, which is the best performing in either cases. In fact, since LB distributes the load on all the available DCs and on different network links, it is more likely for the retrial mechanism to be more effective and reduce the number of rejected requests. On the other hand, MD and ML benefit from the retrial mechanism only at low loads, when blocking is still contained. At higher loads, those strategies tend to fully utilize the resources of few selected DCs (typically the *metro DCs*

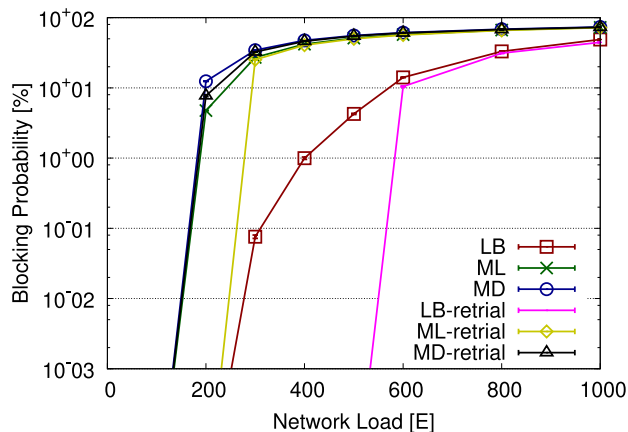


FIGURE 6. Blocking probability - with vs. without retrial mechanism.

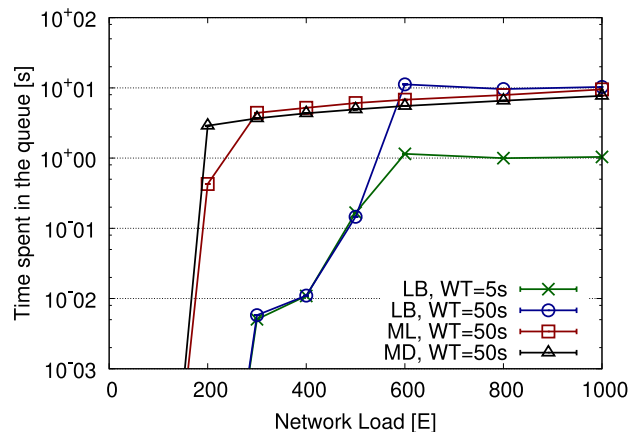


FIGURE 8. Average waiting time of a request before being serviced.

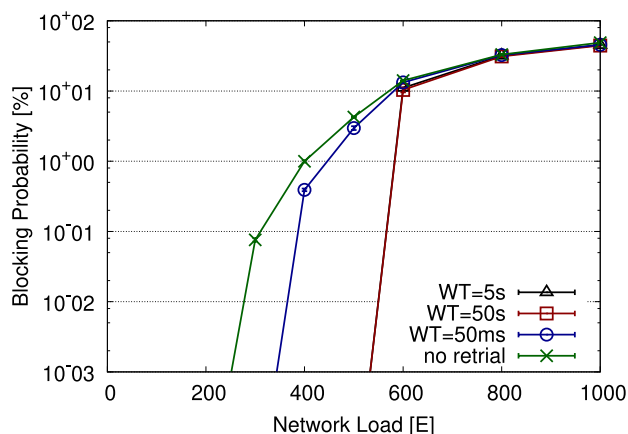


FIGURE 7. LB strategy - blocking probability for different values of WT.

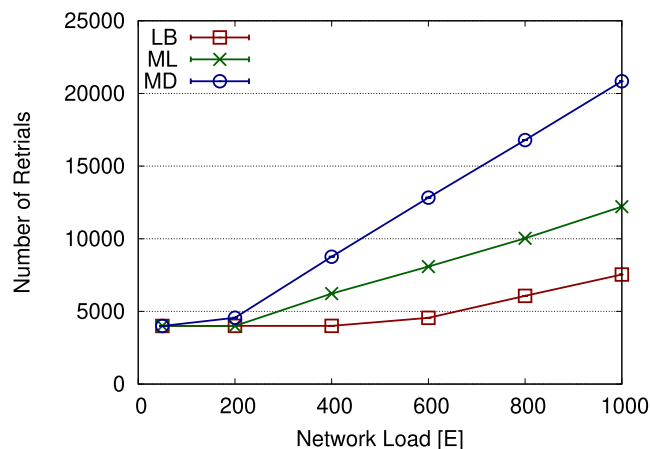


FIGURE 9. Number of retrials - WT = 50s.

which are closer for MD and the *core DCs* which have faster processing time for ML) which strongly limits the effect of the retrial mechanism.

Fig. 7 shows the blocking probability for different values of WT. For the sake of clarity, and considering the results plotted in Fig. 6, only the best-performing strategy (i.e., LB) is analyzed. Results reveal that when increasing WT, the blocking probability of slices deployment decreases. More specifically, we can notice that a low value of WT (e.g., 50ms) has a minor impact on the acceptance rate (around 1% better) especially at high loads since it is more likely that WT expires before resources from other requests are released. On the contrary, for high values of WT, each request has a longer time for being served, increasing the probability of finding available resources and thus improving the acceptance rate. A significant blocking reduction is achieved at medium-low load for WT = 50s. However, such improvement is obtained at the cost of a higher overall slice deployment time.

The time spent in the queue, considering the retrial mechanism, is assessed in Fig. 8 while varying the network load. As previously explained, in order not to reject the request

immediately because of the lack of resources, the request is kept within a queue which introduces an additional delay before deploying the slice. Such delay increases with the increase of the network load since resources become less available and can reach in average tens of seconds as shown in the figure. This raises the necessity for a trade-off between the overall acceptance rate and the time necessary for deploying a slice. Such trade-off mainly depends on the applications requirements in terms of overall delay.

In principle, without considering the resources availability, the time spent in the queue should vary between 0 (i.e., the request is accepted) and WT (i.e., maximum waiting time in the queue after which the request is removed from the queue and is rejected). For comparison purposes, the performance of MD and ML are also reported in Fig. 8. Results show that, LB is the best performing strategy also from the delay point of view. These results are qualitatively similar to the blocking behavior in Fig. 6 since the average waiting time is strictly correlated to the blocking probability. At high loads, the time spent in the queue is almost equal to WT meaning that almost all the queued requests are rejected. In fact, when the network load increases, the arrival rate of the requests

becomes very high with respect to the service time. In such conditions, at the arrival of a new request, it is more likely that the resources already assigned are still occupied and then cannot be released to be reused for the new incoming requests.

Finally, in Fig. 9, we show the number of retrials performed to try to fulfill the slice deployment requests that have been rejected because of the lack of resources. The number of retrials is plotted for the three orchestration strategies as a function of the network load. From the results, we can notice that for low loads (i.e., below 200E), the number of retrials is low (i.e., less than 1% with respect to the number of requests sent) and almost the same for the 3 strategies. In fact, when the network is not congested, it is more likely to find the required resources available as previously demonstrated in Fig. 6 by the low percentage of blocking probability, which decreases the necessity to resort to the retrial mechanism and allows to immediately deploy the slices. By increasing the network load, the number of retrials also increases for the 3 strategies in accordance with the blocking probability rate shown in Fig. 6. The increase is however moderate as for the LB strategy since it is also the strategy that presents the lowest blocking probability percentage thanks to its capability to distribute the load among different data centers and links in the topology. On the contrary, ML and MD strategies reach very high values of blocking percentage at medium and high network loads (i.e., starting from 400E) which is also reflected by the high number of retrials they present. However, although MD and ML have almost the similar values of time spent in the queue, the number of retrials for MD at high loads is greater. In fact, also during the retrial, in the MD strategy the closest DCs are always checked first, which are probably saturated at high loads, thus increasing the number of retrials without improving the performance. ML on the contrary enables a slight redistribution of the load by checking the fastest DCs which, considering the same amount of time spent in the queue, might accept few more requests and then allow for performing less retrials.

C. IMPACT OF NETWORK ROUTING

Figs. 10 and 11 plot the blocking probability when considering the MD and LB strategies, respectively, as a function of the network load while varying the number of shortest paths, i.e., $k = 1, \dots, 4$. For k greater than 1, the paths represent alternative solutions that might be longer than the shortest path but have other advantageous properties (e.g., respond to a higher demand, avoid link failures, avoid specific intermediate nodes). In this paper, we computed the alternative routes by expanding the paths in length order while keeping the same source and destination. Other criteria might be considered in the selection of alternative paths such as calculating the similarity between any two paths or applying weights on previously computed paths.

For the MD strategy (Fig. 10), the sensitivity of the blocking probability to k is negligible except for very low loads. The reason is that MD selects the closest DC, exhausting

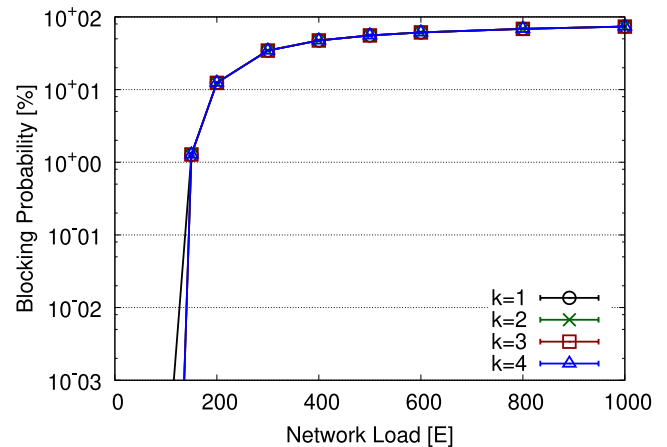


FIGURE 10. Blocking probability vs. load for MD strategy.

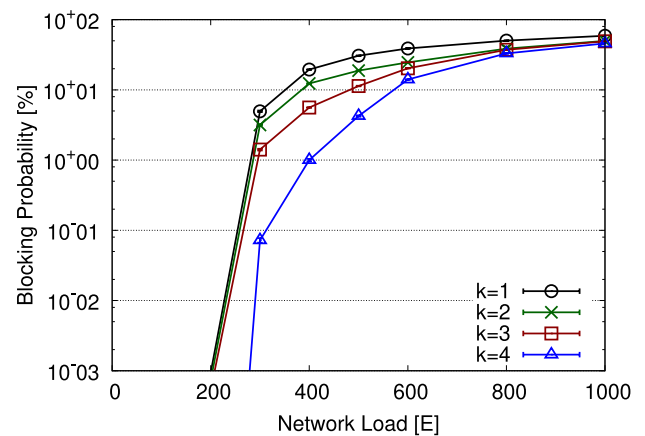


FIGURE 11. Blocking probability vs. load for LB strategy.

quickly its computational resources. Since the main cause of blocking is due to the computational resources, searching for alternative paths has a negligible impact on the overall blocking probability. Instead in the LB strategy (Fig. 11), increasing the pool of candidate paths improves the requests acceptance rate. This improvement occurs because LB strategy searches for unloaded DCs independently from their position. Thus, increasing the value of k augments the probability that at least one of the selected paths has available network resources. At high loads, this improvement diminishes due to overall network congestion.

D. IMPACT OF FLEXI-GRID NETWORK TRANSPARENCY

To compensate for the inter-DC blocking, the use of intermediate OEO conversion can be considered to mitigate the impact of the frequency continuity. More specifically, in an opaque network, only the availability and contiguity of the FSs need to be ensured, and no continuity constraint is enforced contrary to transparent networks. The performance of an opaque network with OEO conversion available in any network nodes is compared to the case of transparent

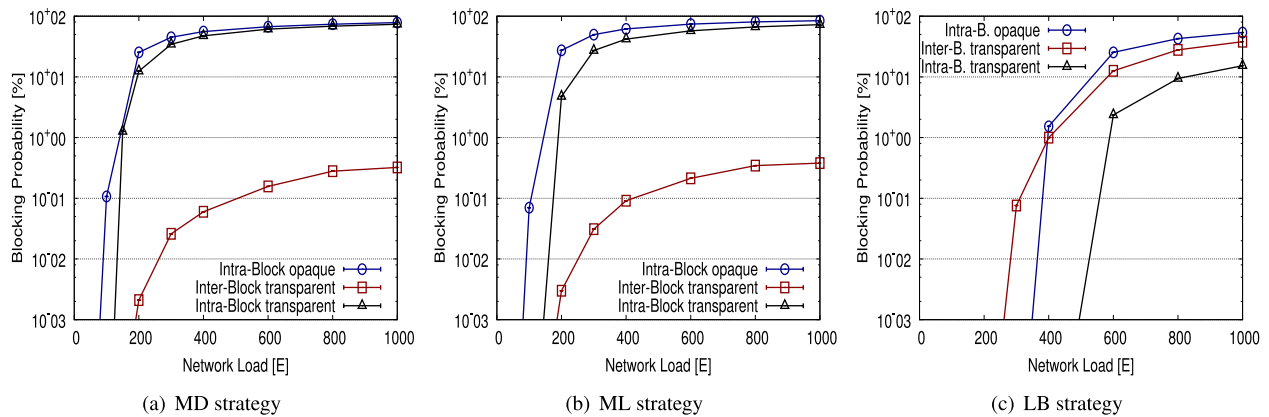


FIGURE 12. Blocking probability components vs. load in opaque and transparent network.

optical networks, previously investigated. The blocking components for the opaque and transparent scenarios are reported in Fig. 12(a)-12(c) for the three orchestration strategies. Results show that contrary to the expectations, in both the MD and ML strategies the opaque scenario presents a higher overall blocking mainly due to the intra-DC blocking component (in both cases, the inter-DC blocking component is null). In fact, given a DC source, these two strategies select always the same DC destination (i.e., the closest and the “quickest” one, respectively) which highly increases the blocking due to the lack of DC resources. Whereas, in the transparent scenario, the continuity constraint slightly increases the inter-DC blocking component thus preventing from allocating new requests. In the LB strategy, a different trend is noticed. Specifically, at low loads, the opaque scenario presents a lower rejection rate than the transparent one, because in LB the inter-DC blocking is the main component and can be lowered by exploiting OEO conversion. At high loads, the DCs are saturated and then the blocking rate in both scenarios becomes almost equivalent mainly because of the increase in the intra-DC blocking.

VI. CONCLUSION

In the upcoming 5G and beyond network infrastructures and in the context of network slicing deployments, the paper compared three orchestration strategies for dynamically selecting resources for VNFs in a scenario of geographically distributed DCs, interconnected by a flexible optical network. Blocking probability and latency are the two critical parameters that were assessed. Simulation results indicated that the load balancing strategy achieves the lowest blocking and contained latency with respect to the ones that minimize the propagation latency and the overall latency. The better performance of the LB strategy can be explained by the more balanced loads across the DCs and thus across also the optical network, leading to a significant reduction of the inter-DC blocking with respect to the other strategies.

To further improve the performance, two approaches aiming at lowering the blocking probability were assessed: a retrial mechanism in the strategy and the removal of

continuity constraints of FSs through OEO conversion in the optical nodes. Results showed that the retrial mechanism is suitable for lowering the blocking at low and medium loads, at the expense of an increased delay for serving the requests. Enabling OEO conversion in the nodes permits also to decrease the blocking probability at low loads, without incurring additional delay, thanks to the possibility to remove the continuity constraints of the FSs. Both approaches are effective in LB. However, the inter-play of the blocking components (inter-DC and intra-DC blocking) and the trade-off between blocking and latency must be carefully evaluated for ensuring the required performance.

As a future work, we plan to replace the simulator by a virtualized testing environment (e.g., FIRE, GENI) to conduct a more realistic performance evaluation. We also envisage to assess the capabilities of other orchestration strategies leveraging, e.g., the proportional allocation criteria.

ACKNOWLEDGMENT

The authors would like to express their special thank to Dr. Isabella Cerutti for her support in this paper work. They are particularly grateful for her valuable suggestion regarding the use of the retrial mechanism as well as her constructive help during the process of performance results analysis.

REFERENCES

- [1] D. Harutyunyan, N. Shahriar, R. Boutaba, and R. Riggio, “Latency-aware service function chain placement in 5G mobile networks,” in *Proc. IEEE Conf. Netw. Softwarization (NetSoft)*, Jun. 2019, pp. 133–141.
- [2] B. Martini, F. Paganelli, P. Cappanera, S. Turchi, and P. Castoldi, “Latency-aware composition of virtual functions in 5G,” in *Proc. 1st IEEE Conf. Netw. Softwarization (NetSoft)*, Apr. 2015, pp. 1–6.
- [3] J. Infuhr and G. R. Raidl, “Introducing the virtual network mapping problem with delay, routing and location constraints,” in *Proc. 5th Int. Conf. Netw. Optim. (INOC)*, Jun. 2011, pp. 105–117.
- [4] H. Hawilo, A. Shami, M. Mirahmadi, and R. Asal, “NFV: State of the art, challenges, and implementation in next generation mobile networks (vEPC),” *IEEE Netw.*, vol. 28, no. 6, pp. 18–26, Nov. 2014.
- [5] A. Hakiri and P. Berthou, “Leveraging SDN for the 5G networks: Trends, prospects and challenges,” Jun. 2015, *arXiv:1506.02876*. [Online]. Available: <http://arxiv.org/abs/1506.02876>
- [6] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, “On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration,” *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1657–1681, 3rd Quart., 2017.

- [7] W. Cerroni, M. Gharbaoui, B. Martini, A. Campi, P. Castoldi, and F. Callegati, "Cross-layer resource orchestration for cloud service delivery: A seamless SDN approach," *Comput. Netw.*, vol. 87, pp. 16–32, Jul. 2015.
- [8] A. Laghrissi and T. Taleb, "A survey on the placement of virtual resources and virtual network functions," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1409–1434, 2nd Quart., 2019.
- [9] A. Rostami, K. Wang, Z. Ghebretensaé, P. Öhlén, and B. Skubic, "First experimental demonstration of orchestration of optical transport, RAN and cloud based on SDN," in *Proc. Opt. Fiber Commun. Conf. Post Deadline Papers*, 2015, pp. 1–3.
- [10] A. Haider, R. Potter, and A. Nakao, "Challenges in resource allocation in network virtualization," in *Proc. 20th ITC Spec. Seminar Netw. Virtualization*, vol. 18, May 2009, p. 20.
- [11] A. Fischer, J. F. Botero, M. T. Beck, H. de Meer, and X. Hesselbach, "Virtual network embedding: A survey," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 4, pp. 1888–1906, 4th Quarter, 2013.
- [12] H. Cao, S. Wu, Y. Hu, Y. Liu, and L. Yang, "A survey of embedding algorithm for virtual network embedding," *China Commun.*, vol. 16, no. 12, pp. 1–33, Dec. 2019.
- [13] M. Pham, D. B. Hoang, and Z. Chaczko, "Congestion-aware and energy-aware virtual network embedding," *IEEE/ACM Trans. Netw.*, vol. 28, no. 1, pp. 210–223, Feb. 2020.
- [14] P. Zhang, H. Yao, M. Li, and Y. Liu, "Virtual network embedding based on modified genetic algorithm," *Peer-Peer Netw. Appl.*, vol. 12, no. 2, pp. 481–492, Mar. 2019.
- [15] X. Zheng, J. Tian, X. Xiao, X. Cui, and X. Yu, "A heuristic survivable virtual network mapping algorithm," *Soft Comput.*, vol. 23, no. 5, pp. 1453–1463, Mar. 2019.
- [16] A. Jahani, L. M. Khanli, M. T. Hagh, and M. A. Badamchizadeh, "EE-CTA: Energy efficient, concurrent and topology-aware virtual network embedding as a multi-objective optimization problem," *Comput. Standards Interfaces*, vol. 66, pp. 103351–103368, Oct. 2019.
- [17] Y. Zong, Y. Ou, A. Hammad, K. Kondepu, R. Nejabati, D. Simeonidou, Y. Liu, and L. Guo, "Location-aware energy efficient virtual network embedding in software-defined optical data center networks," *IEEE J. Opt. Commun. Netw.*, vol. 10, no. 7, pp. 58–70, Jul. 2018.
- [18] M. Zhu, Q. Sun, S. Zhang, P. Gao, B. Chen, and J. Gu, "Energy-aware virtual optical network embedding in sliceable-transponder-enabled elastic optical networks," *IEEE Access*, vol. 7, pp. 41897–41912, 2019.
- [19] T. Gao, X. Li, Y. Wu, W. Zou, S. Huang, M. Tornatore, and B. Mukherjee, "Cost-efficient VNF placement and scheduling in public cloud networks," *IEEE Trans. Commun.*, vol. 68, no. 8, pp. 4946–4959, Aug. 2020.
- [20] P. Cappanera, F. Paganelli, and F. Paradiso, "VNF placement for service chaining in a distributed cloud environment with multiple stakeholders," *Comput. Commun.*, vol. 133, pp. 24–40, Jan. 2019.
- [21] A. Leivadreas, G. Kesidis, M. Ibnkahla, and I. Lambadaris, "VNF placement optimization at the edge and cloud," *Future Internet*, vol. 11, no. 3, p. 69, Mar. 2019.
- [22] S. Agarwal, F. Malandrino, C.-F. Chiasserini, and S. De, "Joint VNF placement and CPU allocation in 5G," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Honolulu, HI, USA, Apr. 2018, pp. 1943–1951.
- [23] B. Martini, M. Gharbaoui, D. Adami, P. Castoldi, and S. Giordano, "Experimenting SDN and cloud orchestration in virtualized testing facilities: Performance results and comparison," *IEEE Trans. Netw. Service Manage.*, vol. 16, no. 3, pp. 965–979, Sep. 2019.
- [24] M. Gharbaoui, B. Martini, D. Adami, S. Giordano, and P. Castoldi, "Cloud and network orchestration in SDN data centers: Design principles and performance evaluation," *Comput. Netw.*, vol. 108, pp. 279–295, Oct. 2016.
- [25] M. Leconte, G. S. Paschos, P. Mertikopoulos, and U. C. Kozat, "A resource allocation framework for network slicing," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2018, pp. 2177–2185.
- [26] J. X. Salvat, L. Zanzi, A. Garcia-Saavedra, V. Sciancalepore, and X. Costa-Perez, "Overbooking network slices through yield-driven end-to-end orchestration," in *Proc. 14th Int. Conf. Emerg. Netw. Exp. Technol.*, Dec. 2018, pp. 353–365.
- [27] *Network Functions Virtualisation (NFV) Release 3; Management and Orchestration; Report on Management and Connectivity for Multi-Site Services*, document GR NFV-IFA 022, ETSI, 2018, vol. 3.1.1.
- [28] ETSI. *Network Functions Virtualisation (NFV); Management and Orchestration*. Accessed: Sep. 24, 2021. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/NFV-MAN/001_099/001/01.01.01_60/gs_NFV-MAN001v010101p.pdf
- [29] OSM. Accessed: Sep. 24, 2021. [Online]. Available: <https://osm.etsi.org/>
- [30] Cloudify. Accessed: Sep. 24, 2021. [Online]. Available: <https://cloudify.co>
- [31] ONAP. Accessed: Sep. 24, 2021. [Online]. Available: <https://www.onap.org/>
- [32] MATILDA. Accessed: Sep. 24, 2021. [Online]. Available: <https://www.matilda-5g.eu/>
- [33] 5G-Transformer. Accessed: Sep. 24, 2021. [Online]. Available: <http://5g-transformer.eu/>
- [34] K. Antevski, J. Martín-Pérez, N. Molner, C. F. Chiasserini, F. Malandrino, P. Frangoudis, A. Ksentini, X. Li, J. SalvatLozano, R. Martínez, I. Pascual, J. Mangues-Bafalluy, J. Baranda, B. Martini, and M. Gharbaoui, "Resource orchestration of 5G transport networks for vertical industries," in *IEEE 29th Annu. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Sep. 2018, pp. 158–163.
- [35] M. Savi, A. Hmaity, G. Verticale, S. Host, and M. Tornatore, "To distribute or not to distribute? Impact of latency on virtual network function distribution at the edge of FMC networks," in *Proc. 18th Int. Conf. Transparent Opt. Netw. (ICTON)*, Jul. 2016, pp. 1–4.
- [36] F. J. Moreno-Muro, C. San-Nicolas-Martinez, M. Garrich, P. Pavon-Marino, O. G. De Dios, and R. L. Da Silva, "Latency-aware optimization of service chain allocation with joint VNF instantiation and SDN metro network control," in *Proc. Eur. Conf. Opt. Commun. (ECOC)*, Sep. 2018, pp. 1–3.
- [37] L. Askari, F. Musumeci, and M. Tornatore, "Latency-aware traffic grooming for dynamic service chaining in metro networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6.
- [38] S. Yang, F. Li, S. Trajanovski, X. Chen, Y. Wang, and X. Fu, "Delay-aware virtual network function placement and routing in edge clouds," *IEEE Trans. Mobile Comput.*, vol. 20, no. 2, pp. 445–459, Feb. 2021.
- [39] F. Lin, B. Chen, M. Dai, G. Su, and R. Lin, "VNF placement for service chaining in sliceable-transceiver-equipped IP over EONs," in *Proc. Asia Commun. Photon. Conf. (ACPC)*, 2019, pp. 1–3.
- [40] Y. Li, Y. Zhao, B. Li, X. Yu, H. Yang, X. Wang, and J. Zhang, "Joint balancing of IT and spectrum resources for selecting virtualized network function in inter-datacenter elastic optical networks," *Opt. Exp.*, vol. 27, no. 11, pp. 15116–15128, May 2019.
- [41] L. Ruiz, R. J. Duran, I. de Miguel, N. Merayo, J. C. Aguado, P. Fernandez, R. M. Lorenzo, and E. J. Abril, "Joint VNF-provisioning and virtual topology design in 5G optical metro networks," in *Proc. 21st Int. Conf. Transparent Opt. Netw. (ICTON)*, Jul. 2019, pp. 1–4.
- [42] L. Askari, F. Musumeci, and M. Tornatore, "A techno-economic evaluation of VNF placement strategies in optical metro networks," in *Proc. 4th Int. Conf. Comput., Commun. Secur. (ICCCS)*, Oct. 2019, pp. 1–8.
- [43] T. Phung-Duc, "Retrial queueing models: A survey on theory and applications," in *Stochastic Operations Research in Business and Industry*. Singapore: World Scientific, 2017.
- [44] T. Phung-Duc and K. Kawanishi, "An efficient method for performance analysis of blended call centers with redial," *Asia-Pacific J. Oper. Res.*, vol. 31, no. 2, pp. 665–716, 2014.
- [45] J. Wang and W. W. Li, "Noncooperative and cooperative joining strategies in cognitive radio networks with random access," *IEEE Trans. Veh. Technol.*, vol. 65, no. 7, pp. 5624–5636, Jul. 2016.
- [46] T. V. Do, "Solution for a retrial queueing problem in cellular networks with the fractional guard channel policy," *Math. Comput. Model.*, vol. 53, nos. 11–12, pp. 2059–2066, Jun. 2011.
- [47] M. Gharbaoui, I. Cerutti, B. Martini, and P. Castoldi, "An orchestrator of network and cloud resources for dynamic provisioning of mobile virtual network functions," in *Proc. IEEE NetSoft Conf. Workshops (NetSoft)*, Jun. 2016, pp. 98–101.
- [48] B. Martini, M. Gharbaoui, I. Cerutti, and P. Castoldi, "Network and datacenter resource orchestration strategies for mobile virtual networks over telco clouds," in *Proc. 18th Int. Conf. Transparent Opt. Netw. (ICTON)*, Jul. 2016, pp. 1–4.
- [49] S. Fichera, R. Martínez, B. Martini, M. Gharbaoui, R. Casellas, R. Vilalta, R. Muñoz, and P. Castoldi, "Latency-aware resource orchestration in SDN-based packet over optical flexi-grid transport networks," *J. Opt. Commun. Netw.*, vol. 11, no. 4, pp. B83–B96, Apr. 2019.
- [50] X. Wang, M. Razo, M. Tacca, N. So, and A. Fumagalli, "Five resource allocation strategies for the cloud infrastructure," in *Proc. 17th Int. Conf. Opt. Netw. Design Modeling (ONDM)*, Apr. 2013, pp. 13–18.
- [51] B. Martini, M. Gharbaoui, and P. Castoldi, "Cross-functional resource orchestration in optical telco clouds," in *Proc. 17th Int. Conf. Transparent Opt. Netw. (ICTON)*, Jul. 2015, pp. 1–5.
- [52] P. Öhlén, B. Skubic, A. Rostami, M. Fiorani, P. Monti, Z. Ghebretensaé, J. Mártensson, K. Wang, and L. Wosinska, "Data plane and control architectures for 5G transport networks," *J. Lightw. Technol.*, vol. 34, no. 6, pp. 1501–1508, Mar. 15, 2016.

- [53] L. Velasco, A. Asensio, A. Castro, J. Berral, D. Carrera, V. López, and J. Fernández-palacios, "Cross-stratum orchestration and flexgrid optical networks for data center federations," *IEEE Netw.*, vol. 27, no. 6, pp. 23–30, Nov. 2013.
- [54] R. Martinez, F. Cugini, R. Casellas, F. Paolucci, R. Vilalta, P. Castoldi, and R. Munoz, "Control plane solutions for sliceable bandwidth transceiver configuration in flexi-grid optical networks," *IEEE Commun. Mag.*, vol. 54, no. 8, pp. 126–135, Aug. 2016.
- [55] L. Velasco, M. Ruiz, A. Castro, O. Pedrola, M. Klinkowski, D. Careglio, and J. Comellas, "On the performance of flexgrid-based optical networks," in *Proc. 14th Int. Conf. Transparent Opt. Netw. (ICTON)*, Jul. 2012, pp. 1–4.
- [56] M. Klinkowski, M. Ruiz, L. Velasco, D. Careglio, V. Lopez, and J. Comellas, "Elastic spectrum allocation for time-varying traffic in flex-grid optical networks," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 1, pp. 26–38, Jan. 2013.
- [57] J. Zhang, Y. Zhao, X. Yu, J. Zhang, M. Song, Y. Ji, and B. Mukherjee, "Energy-efficient traffic grooming in sliceable-transponder-equipped IP-over-elastic optical networks [invited]," *J. Opt. Commun. Netw.*, vol. 7, no. 1, p. A142, Jan. 2015.
- [58] J.-F. Labourdette and Z. Zhang, "Opaque and transparent networking," *Opt. Netw. Mag.*, Jan. 2003. [Online]. Available: https://www.researchgate.net/publication/242376315_Opaque_and_Transparent_Networking_Tutorial_Corner
- [59] N. Cvijetic, A. Tanaka, P. N. Ji, K. Sethuraman, S. Murakami, and T. Wang, "SDN and OpenFlow for dynamic flex-grid optical access and aggregation networks," *J. Lightw. Technol.*, vol. 32, no. 4, pp. 864–870, Feb. 15, 2014.
- [60] I. Tomkos, S. Azodolmolky, J. Solé-Pareta, D. Careglio, and E. Palkopoulou, "A tutorial on the flexible optical networking paradigm: State of the art, trends, and research challenges," *Proc. IEEE*, vol. 102, no. 9, pp. 1317–1337, Sep. 2014.
- [61] W. Shi, Z. Zhu, M. Zhang, and N. Ansari, "On the effect of bandwidth fragmentation on blocking probability in elastic optical networks," *IEEE Trans. Commun.*, vol. 61, no. 7, pp. 2970–2978, Jul. 2013.
- [62] B. C. Chatterjee, S. Ba, and E. Oki, "Fragmentation problems and management approaches in elastic optical networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 183–210, 1st Quart., 2018.
- [63] S. Fichera, B. Martini, R. Martínez, M. Gharbaoui, R. Casellas, R. Vilalta, R. Muñoz, and P. Castoldi, "Experimental evaluation of dynamic resource orchestration in multi-layer (packet over flexi-grid optical) networks," *Photon. Netw. Commun.*, vol. 40, no. 3, pp. 137–148, Dec. 2020.
- [64] A. Pagès, J. Perelló, and S. Spadaro, "Lightpath fragmentation for efficient spectrum utilization in dynamic elastic optical networks," in *Proc. 16th Int. Conf. Opt. Netw. Design Modelling (ONDM)*, Apr. 2012, pp. 1–6.
- [65] J. Kim and B. Kim, "A survey of retrial queueing systems," *Ann. Oper. Res.*, vol. 247, no. 1, pp. 3–36, Dec. 2016.
- [66] J. Y. Yen, "Finding the K-shortest loopless paths in a network," *Manage. Sci.*, vol. 17, no. 11, pp. 712–716, Jul. 1971.
- [67] M. Gharbaoui. (2021). *Resource Orchestration Strategies With Retrials*. [Online]. Available: <https://github.com/mgharb/Orchestration-DCs-Retrials>
- [68] A. Betker et al., "Reference transport network scenarios," MultiTeraNet Rep., Jul. 2003. Accessed: Sep. 24, 2021. [Online]. Available: https://www.ikr.uni-stuttgart.de/documents/simlib/Referenz_Netze_v14_full.pdf
- [69] T. Nandagopal and K. P. N. Puttaswamy, "Lowering inter-datacenter bandwidth costs via bulk data scheduling," in *Proc. 12th IEEE/ACM Int. Symp. Cluster, Cloud Grid Comput. (CCGRID)*, May 2012, pp. 244–251.



B. MARTINI is currently the Head of research with the CNIT National Laboratory of Photonic Networks and Technologies (PNT Lab) and an Affiliate Researcher with Sant'Anna School of Advanced Studies, Italy. Before joining CNIT PNT Lab, she worked for two large telco companies, Italtel and Marconi Communications (currently Ericsson). She is an Adjunct Professor with Sant'Anna School of Advanced Studies and the University of Pisa, Italy. She has been involved in several national/EU research projects, the recent ones 5GPPP 5GEX, 5GTRANSFORMER, and 5GROWTH, and in several FIRE projects (OFELIA, FED4FIRE+, TRIANGLE, and 5GINFIRE) with leading roles. She has coauthored more than 100 articles in international journals and conference proceedings. Her research interests include network virtualization and orchestration in SDN/NFV/5G environments, service platforms for next-generation networks, network control/management architectures, and security solutions for multidomain IP/optical networks and NFV deployments.



G. CECCHETTI received the master's degree in computer engineering from the University of Pisa, the Ph.D. degree in innovative technologies from Scuola Superiore Sant'anna, and the Ph.D. degree in clinical engineering from the University of Trieste. He is currently a Researcher at TeCIP Institute. His skills and activities are in the ICT sector, where he has gained many years of experience in the field of computer architecture, digital systems, data centers, and embedded systems. He has collaborated on many national and European research projects. He is a Technical Leader of the Communication Platform for Traffic Management Demonstrator (OPTIMA) Project funded by Shift2Rail as part of the H2020 Program. His teaching activity is in the field of computer architecture and digital systems. He has published articles in international journals and conference proceedings. His research interests include the support of quality of service on heterogeneous networks, IEEE 802.11e networks, railway signaling, and communication systems. He has worked as a reviewer for numerous international journals and conferences.

laborated on many national and European research projects. He is a Technical Leader of the Communication Platform for Traffic Management Demonstrator (OPTIMA) Project funded by Shift2Rail as part of the H2020 Program. His teaching activity is in the field of computer architecture and digital systems. He has published articles in international journals and conference proceedings. His research interests include the support of quality of service on heterogeneous networks, IEEE 802.11e networks, railway signaling, and communication systems. He has worked as a reviewer for numerous international journals and conferences.



M. GHARBAOUI received the Ph.D. degree in innovative technologies of information and communications engineering, and robotics from Scuola Superiore Sant'anna, Pisa, Italy, in 2012. She is currently a Technologist at Scuola Superiore Sant'anna. She has been involved in several EU projects. She has coauthored more than 50 papers appeared in international journals and conferences. Her main research interests include software-defined networking, network function virtualization, network orchestration, the development and the implementation of service-oriented architectures, and service management for smart cities.

virtualization, network orchestration, the development and the implementation of service-oriented architectures, and service management for smart cities.



P. CASTOLDI (Senior Member, IEEE) was at Princeton University, USA, overall about two years, in 1996, 1997, 1999, and 2000. He has been a Professor at Scuola Superiore Sant'Anna, Pisa, Italy, since 2001. He is currently a Leader of the "Networks and Services" research area at Scuola Superiore Sant'Anna. He is the author of more than 400 international publications. His research interests include telecommunications networks and system both wired and wireless, and more recently reliability, switching paradigms, and control of optical networks, including application-network cooperation mechanisms for cloud networking.