POLITECNICO DI TORINO
Repository ISTITUZIONALE

A unifying orchestration operating platform for 5G

(Article begins on next page)

20 May 2024

A Unifying Orchestration Operating Platform for 5G

Antonio Manzalini[1], Marco Di Girolamo[2], Giuseppe Celozzi[3], Fulvio Bruno[3], Giuliana Carullo[4], Marco Tambasco[4], Gino Carrozzo[5], Fulvio Risso[6], and Gabriele Castellano[6]

[1] Telecom Italia Mobile, Italy
[2] HPE, Italy
[3] Ericsson, Italy
[4] CoRiTeL, Italy
[5] Nextworks, Italy
[6] Politecnico di Torino, Italy

Abstract. 5G will revolutionize the way ICT and Telecommunications infrastructures work. Indeed, businesses can greatly benefit from innovation introduced by 5G and exploit the new deep integration between ICT and networking capabilities to generate new value-added services. Although a plethora of solutions for virtual resources and infrastructures management and orchestration already exists (e.g., OpenDaylight, ONOS, OpenStack, Apache Mesos, Open Source MANO, Docker Swarm, LXD/LXC, etc.), they are still not properly integrated to match the 5G requirements. In this paper, we present the 5G Operating Platform (5G-OP) which has been conceived to fill in this gap and integrate management, control and orchestration of computing, storage and networking resources down to the end-user devices and terminals (e.g., smart phone, machines, robots, drones, autonomous vehicles, etc). The 5G-OP is an overarching framework capable to provide agnostic interfaces and a universal set of abstractions in order to implement seamless 5G infrastructure control and orchestration. The functional structure of the 5G-OP, including the horizontal and vertical interworking of functions in it, has been designed to allow Network Operators and Service Providers to exploit diverse roles and business strategies. Moreover, the functional decoupling of the 5G-OP from the underneath management, control and orchestration solutions allows pursuing faster innovation cycles, being ready for the emergence of new service models.

Keywords: 5G, SDN, NFV, Cloud, Management, Control, Orchestration, API

1 Introduction

In the era of massive explosion of pervasive powerful devices and the high diffusion of fixed and mobile ultra-broadband, both ICT and Telecommunications infrastructures are evolving towards 5G. A systemic trend, usually termed as

network Softwarization, is in place, where ICT and Telecommunication infrastructures are radically leveraging on software decoupling and virtualization technologies. In this context, Cloud, Edge and Fog Computing [1], Network Function Virtualization (NFV) [2] and Software Defined Networking (SDN) [3] are the most important enabling technology frameworks, which can be seen as different dimensions of an overall effort to implement the 5G paradigm. Indeed, the design of 5G infrastructure is looking much beyond a simple evolution of current 4G/LTE mobile networks. 5G will not only provide higher radio bit rates or reduced end-to-end latency: it will be based on a true fixed-mobile convergence and a deeper integration of IT and networking capabilities. Eventually, the overall 5G infrastructure will be simplified, for example, by reducing the number of layers and overcoming the need for silos; 5G functionalities will be virtualized and dynamically allocated onto horizontal network-service platforms, pervasively deployed, including the access segments (e.g., SoftRAN, CloudRAN [4]). This will improve the levels of pervasivity, dynamicity, programmability and robustness of 5G infrastructures, whilst increasing the degrees of complexity, volatility and unpredictability. Indeed, human-made operations will not be able to face the emerging control and management challenges, especially in highly dynamic environments, highlighting - as a consequence - the urgent need to redesign the processes for infrastructure control, management and orchestration, namely by introducing higher levels of automation. The provisioning of 5G end-to-end services, allocated and executed in network slices, will require orchestration capabilities, a universal set of abstractions and standard Open APIs. By filling this gap, it will be possible to achieve network services separation as well as proper isolation required to ensure both service quality and security. Multiple levels of orchestration are needed for service and resource (network and computing) provisioning, to ensure that the network slices deliver services at the required quality levels and with optimal resource utilization. Furthermore, a policy-based control can be used to validate the performance of network slices, and more widely to enact service monitoring. The first concrete exploitations of 5G Software Networks implementing the functionalities introduced above are expected to be rolled-out by 2020.

Within this research framework, the authors of this paper have elaborated a reference functional architecture for a unifying Operating Platform for 5G (5G-OP), which integrates management, control and orchestration of computing, storage and networking resources down to the end-user devices and terminals (e.g., smart phone, machines, robots, drones, autonomous vehicles, etc). Despite numerous research and development efforts in the area of SDN and NFV have been going on for many years, with a number of products now in the market and a significant steering role of large-scale open source development communities (e.g. those behind the developments of OpenDaylight [5], ONOS [6], OpenStack [7, 8], Apache Mesos [9], OpenSource MANO [10], Docker Swarm, LXD/LXC, etc.), no consolidated control and orchestration solutions exist. For instance, SDN controllers lack common application interfaces (northbound Interfaces), NFV orchestrators rely on different infrastructure models, etc. The heterogeneity of

the implemented solutions is heading for complex and costly situations, with high fragmentation, which in turn creates uncertainty and the risk of delaying 5G innovations.

5G-OP originates from the observation of the state of the art, to address this issue, by providing a platform that can easily taken up by Telcos and service providers to implement in an end-to-end way the various 5G scenarios for their vertical customers. The main innovation of the 5G-OP lies in its overarching characteristics. 5G-OP does not intend to develop one more control-orchestration platform to replace the ones in current state of the art (OpenDaylight, ONOS, CORD [11], M-CORD [12], OpenStack, etc.). Contrarily, 5G-OP sits on top of them, exposing orchestrated services across multiple different network and non-network domains through unified interfaces, providing a set of abstractions and adaptation functions. The agnostic and overarching characteristics of the 5G-OP allows true decoupling from the underneath control-orchestration platforms, which thus becomes pluggable into the 5G-OP. Therefore, 5G-OP does not add another layer of complexity; contrarily, it is radically simplifying the integration of current and future platforms in the 5G-OP in a way to provide end-to-end services spanning across heterogeneous technological infrastructures (e.g., cloud, network, fog, terminals), and that can evolve over time.

The outline of the paper is the following. Section 2 presents the master design guidelines of the 5G-OP, while Section 3 reports some of the possible application scenarios in which the 5G-OP can definitely impact and create value. A brief description of a preliminary prototype is presented in Section 4, while Section 5 provides some results from the experimental validation we performed. Finally, Section 6 reports our closing remarks and sketches some future evolutions of the 5G-OP.

2   The 5G Operating Platform

The 5G-OP is an overarching architecture, with agnostic interfaces and a rich set of abstractions, offering seamless integration of current and future infrastructure management, control and orchestration solutions (e.g., ONOS, OpenDaylight, OpenStack and even Robot Operating System [13], etc.). In 5G-OP, each service orchestrator exposes its resources and services via a Northbound API (NBI) to a consumer and a Southbound API (SBI) to use resources and services from underlying serving orchestrators. In order to keep the overall control architecture manageable across multiple distributed and chained instances, it is fundamental to design Northbound APIs that can abstract the various primitive functions of each Orchestrator instance, and that can allow the formulation of the typical actions on resources and services like composition, chaining and so on, in the form of "intents". Interfaces are defined towards the Infrastructure controllers (Southbound APIs) and intent-based APIs are defined towards the Application/tenant (Northbound API). NBI API is meant to make this available to the verticals and Over-The-Top (OTT), that will use them to implement their applications.

The technical approach we propose aims at enabling a flexible coupling with the underlying infrastructure control and orchestration platforms. In particular, we focus on the aspects that are involved when an application wants to make use of 5G-OP and leverage the best of the underlying infrastructure, which translates in the possibility to exploit the peculiar characteristics available in each domain. This implies that an overarching orchestrator based on the *minimum common denominator* is not appropriate, as it hides possible unique features available in some portions of the infrastructure. Instead, the 5G-OP must export the above capabilities to the upper layers in order to allow potential usage of them.

Open APIs are a key aspect of this approach that guarantees an easy integration of infrastructure platforms, thus allowing the boost and quick exploitation of open innovations creation of vertical application services on 5G. In other words, the core of the 5G-OP consists in an overarching orchestration framework operating over different 5G infrastructures including core/edge/access networks, cloud, and end-user devices and terminals, while leveraging unified service and infrastructure abstractions to support end-to-end orchestration across different technology domains as well as generalized orchestration of infrastructure and application services.

2.1   Unified service model and 5G abstractions

5G-OP seamlessly supports new capabilities (e.g., a new type of network infrastructure) and services (e.g., application-specific orchestrators, or a module that can handle a new type of IoT sensor). 5G-OP offers allows new capabilities and adds support for new technological domains and various administrative domains (i.e., with technology agnostic and business federation mechanisms) by a plug-and-play approach. This way, new services and applications can easily access and use such new features.

2.2   A Generalized Orchestration Space

With the concept of the "shared orchestration space" (shown in Figure 1), the 5G-OP brings together two problems that are usually considered separately: infrastructure-level and application-layer orchestration. A generalized orchestration workflow/process should be devised that involves the composition of both application and infrastructural resources, capabilities and services while adapting the composite services to the different and/or ever-changing contextual information [14].

5G-OP aims to achieve a model-driven provisioning of services through the different levels of orchestration. Thus one of its key features is the availability of a common data model based on graphs correlating and merging together services, resources and capabilities to represent relationships and workflows, as shown in the example in Figure 2. The relations between orchestration modules can be defined as a set of transformations and be formally verified. In order to improve performance and scalability of services, the transition across multiple layers can be optimized through an equivalent and formally verified graph-based model,
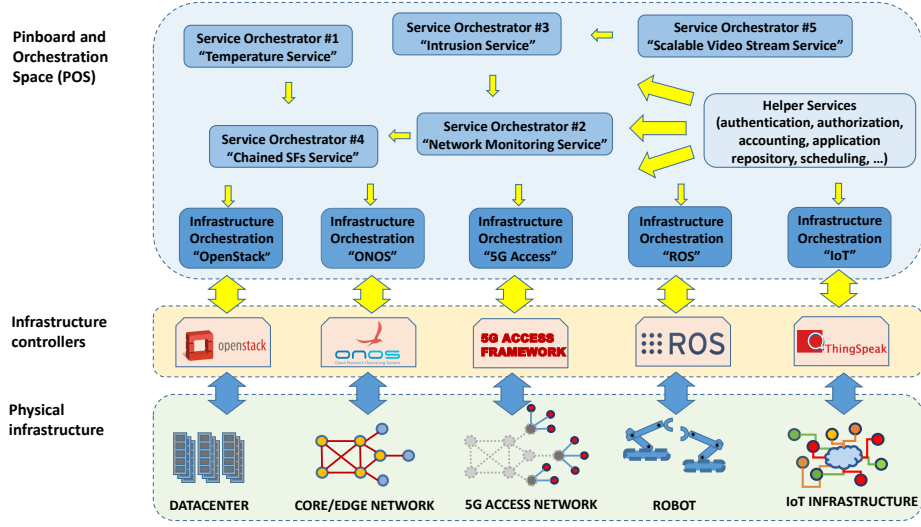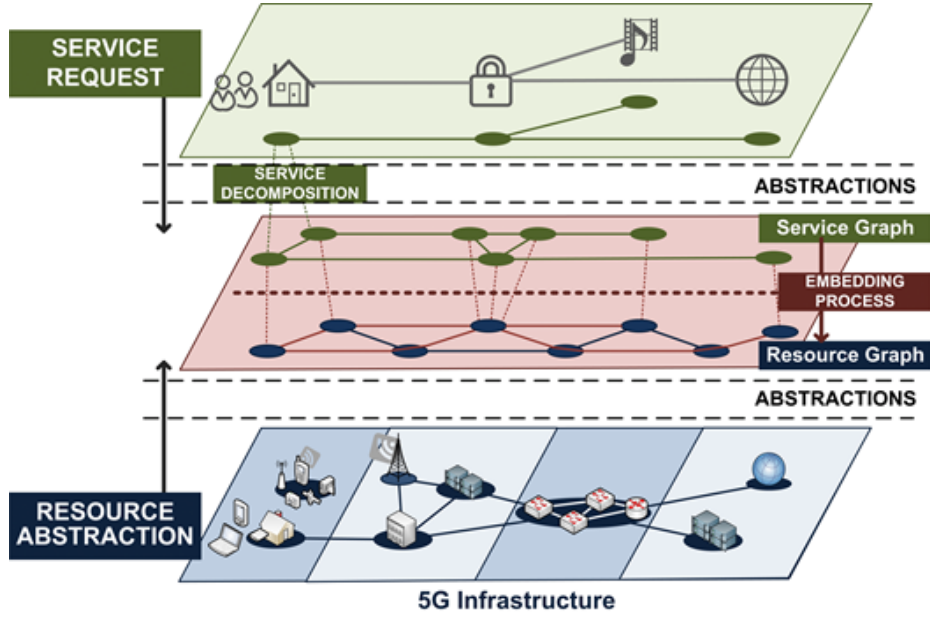
Fig. 1. 5G-OP architecture.

Fig. 2. Service and resource graphs.

enabling the definition of an allowed set of transformations. The API exposed by this consistent approach between design and runtime phases will enable 5G-

OP to reduce capacity churn, eliminating isolated under/unused capacity and delivering the best user experience.

2.3 Architectural principles

5G-OP allows a generalized, flexible and de-structured orchestration workflow in which orchestrators can decompose a service request into more elementary ones. The generalized orchestration is assured by proper abstractions and interfaces offered by orchestrators while interacting each other to address service requests in a structured service producer-consumer relationship. In 5G-OP, each service orchestrator exposes a Provider API for the northbound interface and a Consumer API for the southbound interface. Composition is achieved by attaching a Provider to a Consumer, thus providing the additional advantage of allowing horizontal composition, not requiring strict vertical hierarchies. At the Provider API, different logical views of the underlying resource and service capabilities (for network and non-network components) are provided to the service consumers, thus realizing the slicing concept.

The 5G-OP Provider API supports the concept of intents to ease the way a service consumer can request a service from the underlying layer, ignoring technological details on how the actual resources are configured and how the service is provisioned. At the Consumer API, abstraction is mainly aimed at wrapping details of different devices and resources in the underlying layer, controlled as objects with generalized capabilities across various technology domains.

2.4 5G-OP and new benchmarking principles

Benchmarking is important both in selecting and purchasing components and solutions but even more for planning, deployment and in-operation optimization where knowledge about performance benchmarks is non-negligible. We have to go beyond designing traffic traces for benchmarking of one virtualized node or network function (e.g., as suggested in [15]) and consider the definition of complex usage scenarios encompassing a mix of services across multiple carrier-grade data centers instances and controllers. Ultimate goal is benchmarking and comparing deployment options for different sets of network applications and their combined usage patterns, as well as taking into account different scaling scenarios. As a result, it will be possible to obtain performance knowledge which is comparable across - and from - different sources and cases to better appraise the resource footprint and performance achievements, and to detect the most efficient algorithms for mapping services to resources. Advances in benchmarking definitions, methods and supporting tools, and the corresponding standardization (see for instance [16]) will be greatly beneficial to the 5G-OP vision. Furthermore, the 5G-OP architectural principles are also important for enabling such advanced benchmarking scenarios.

6

2.5 5G-OP security perspective

While 5G-OP provides significant advantages in terms of flexibility, automation and efficiency of network controls, possible security threats need to be successfully addressed and mitigated by enforcing protection measures. In particular, since services, SDN controllers and data are hosted in the cloud, the infrastructure itself can offer a potential target to attacks. For example, an SDN controller in the cloud may not only be compromised via security breaches in the virtualization layer, but also via Application Programming Interfaces (APIs) exposed to overarching applications. As a result, APIs are a topic of great interest - in general - in the information security community: they are still affected by security issues, often caused by poorly written code that can quickly become dangerous, especially when it involves Open APIs. Insecure Open APIs come with major risks for both applications and underlying data. 5G-OP will consider this issues and will thus apply, when possible, best practices to deliver secure APIs.

More in general, 5G-OP will exploit both scalable allocation of resources and the programmability provided by SDN to enable the deployment of security solutions in a flexible and efficient manner.

3 5G Application Scenarios

In this Section we propose some key application scenarios in which 5G-OP would greatly impact control and orchestration.

3.1 5G enabling Cloud Robotics and Industry 4.0

The development of increasingly complex cognitive capabilities spanning across the 5G infrastructure up to the terminals (e.g., robots, drones or autonomous machines), will offer benefits in terms of automated operational processes and optimized costs, as well as enabling the development of new service scenarios. This will pose challenging requirements for ensuring ultra-low latencies in "closing" the "interaction loop" made of sensors, processing-storage-networking nodes, systems and actuators. 5G for Cloud robotics offers several examples of use cases for Industry 4.0 and Precision Agriculture. Moreover, it is likely that remotely controlled and operated robots will enable remote surgery and will open up a new world of domestic applications. The benefits of a 5G-OP overarching orchestration framework in this context are quite important, since 5G-OP allows to control the end-to-end service provisioning and lifecycle across the various network and non-network domains while matching the challenging KPIs on latency, bandwidth, mobility and fast adapting service chain.

3.2 Media Content Everywhere and Home services

Delivery of content is subject to a massive growth, due to a variety of devices and the demand for increased video quality in terms of resolution and frame

rate. To this end, cloud offers the possibility to increase the amount of contents to be stored, but Live and On-demand content services need to be delivered as if they were stored locally and that is where a Content Delivery Network (CDN) is needed. Traditional CDN cache nodes offered as dedicated physical appliances or software with standard but dedicated hardware, come with disadvantages like the capacity of the devices that needs to be designed for peak hours on weekends, while on weekdays and business hours, they are almost unused. Furthermore, they do not scale well to deal with unforeseen capacity needs (e.g., live events). Integrating nodes of Content Delivery Networks via NFV orchestration into operator networks can be the most effective and cost-efficient way to answer the challenges imposed by Video Traffic Delivery scenario. Content streams will steam out of compute/storage nodes nearer to the end customer, saving upper network links and equipment and delivering, at the same time, higher bandwidth and quality.

Moreover, the availability of high bandwidth access and NFV technology can facilitate the virtualization of the home environment, built on top of simple, low cost and low maintenance devices at the customer premises. NFV technologies are ideal to support this shift of computational workload from formerly dispersed functions with minimal cost introduced on a grow-as-you-need basis. Even more important, 5G-OP could seamlessly integrate the home environment, with the foreseen IoT-oriented services, into the whole world of services delivered by the Telco provider. This can enable the seamless deployment of virtual appliances and services (e.g., analytics) on a virtual platform that can take care of decomposing the requested service (and the associated data) and relocating its components on the best location based on service and user constraints.

3.3 Remote control of critical operations

The possible scenarios of remote operations in industry are numerous, and each scenario brings its unique set of challenges. Power plants, mines, construction sites, and oil platforms can be target environments for remote control, which will reduce personnel exposure to an abundance of risks associated including, for example, the presence of heavy machinery and chemicals. Remote operation solutions allow people to operate machinery from a control centre at another site. Anyway, the distance between these sites puts challenges in terms of both delay and connection reliability that can be solved with proper orchestrated solutions.

In order to be as productive as on-site manual operators, it is needed to reproduce the sense of a surrounding environment, including all the real-time feedback that we can obtain when the operator is physically located in the (dangerous) environment.

The requirements set by some use cases, which are of interest of both society and industries, cannot easily be met by existing communication technologies. Instead, remote operation solutions with 5G connectivity must be used, which can offer a number of benefits as it can provide connectivity to mobile machinery and devices. Furthermore, orchestration of 5G slices can deliver high reliability, and the required level of security.

4   5G-OP prototype

A preliminary version of our 5G-OP has been implemented in the FROGv4 open-source prototype [17]. It includes six modules: a main orchestrator (called *telco orchestrator*), three *domain orchestrators* each one supporting a different technological infrastructure, and two *helper services*. Our prototype addresses the requirement from Telco operators to split their network infrastructure across different domains, either because of technical diversities (i.e, a data center requires a different controller compared to a transport SDN network) or scalability reasons. Thus, we designed the Telco orchestrator as the entry point for any service request targeting the infrastructure. Upon an incoming service request from the northbound interface, the Telco orchestrator is in charge to determine how the service is mapped on the available domains based on the capabilities exposed by each domain controller. If needed, the service request will be broken down into smaller requests targeting individual domains (e.g., a set of VMs to be allocated in the data center, a set of "virtual wires" established in a SDN domain), which are then sent to the selected domain controllers. In addition, the Telco orchestrator will create the proper logical connections (e.g., by setting up point-to-point VLANs, IPsec/GRE tunnels, etc.) that allow the portion of the service instantiated in a first domain to deliver the traffic to the following network functions, installed in a different domain. The Telco orchestrator interacts with three domain orchestrators. These orchestrators provide the interface between the Pinboard and Orchestration Space (POS) and technology-specific individual domain controllers, which will be presented with more details in the next sections. Finally, we designed two helper services: one that is dedicated to authentication and security purposes, the other that provides storage for all the above modules.

These six modules constitute the first implementation of a 5G-OP. Each module exposes a northbound interface that includes both a REST API for direct service invocation and an access to a message bus, where publish/subscribe primitives are used. The message bus is implemented with DoubleDecker [18], a ZeroMQ based distributed message solution, that provides a hierarchical brokering system. It allows different modules to be notified when a given event happens (*subscribe* primitive) and enables a module to send a given information to an unknown set of consumers (*publish* primitive). This feature, coupled with the YANG-derived [19] common data module across all the services, creates a common ground between modules that allows to produce and consume data, and to interact each other in a seamless way.

Each domain orchestrator has been implemented as a software module that receives service requests through its northbound interface, and that interacts (e.g., to deploy NFs, create network paths, retrieve information on available services and network topology) with the technology-specific controller (e.g., OpenStack or ONOS) of the underlying domain through the controller-specific APIs. Thus, the prototype can leverage existing vanilla infrastructure controllers, which can be integrated in our orchestration framework by simply writing the proper domain orchestrator on top of them. In the following we describe the three do-

main orchestrators we developed: the SDN domain orchestrator, the Openstack domain orchestrator and the Universal Node domain orchestrator.

4.1 SDN domain orchestrator

The *SDN domain orchestrator* sits on top of the ONOS (Falcon release) controller and aims at supporting SDN domains. This orchestrator queries the ONOS controller to retrieve the underlying network topology (consisting of OpenFlow switches) and to detect all the available software bundles. The former is exported as unique "big switch" through the orchestrator northbound interface, while the latter are advertised as potential services that can be turned on upon request. Currently, the list of available actions is limited to the possibility to (*i*) control software switches (e.g., creating logical switching instances, adding/removing ports), (*ii*) handle network paths (e.g., creating, redirecting, and tearing down a path between two ports of the "big switch"), and (*iii*) start one of the advertised service[1]. Network paths are implemented by installing OpenFlow rules in any hardware/software switch; however, this is considered an internal detail and hence it is not exported to the POS, which can control the infrastructure with more abstract primitives such as end-to-end (logical) paths.

When the SDN orchestrator receives a service request from its northbound interface, it calls the proper set of ONOS functions in order to set up the required paths and, if needed, starts the requested application (i.e., software bundle). Moreover, the SDN domain orchestrator receives also the precise ingress/egress traffic characterization of the endpoints that are part of the service graph, which are used, between all, to recognize where the traffic comes from (and the possible encapsulation, e.g., in a specific VLAN ID) and handle it accordingly. In this respect, the SDN orchestrator has to inject the proper flow-rules in the underlying OpenFlow domain border switches, which are used to properly tag/untag packets entering in (and exiting from) the domain as indicated by the endpoints.

Finally, since ONOS does not support the parallel execution of multiple instances of the same bundle, applications must also manage multi-tenancy by themselves in order to support service chains deployed at the same time and requiring the same bundle.

4.2 OpenStack domain orchestrator

This domain orchestrator is responsible for the deployment of service requests in OpenStack-based data centers, solving the necessity to handle services that spans across multiple domain. In particular, it enables a complete control on the traffic that enters in (or exits from) the considered domain. As an example, a service chain split across two domains may require the setup of a GRE tunnel between two boundary interfaces of the involved domains. This behavior is

---

[1] In SDN networks, the creation of a chain of applications is not trivial; hence, the current prototype supports only service requests that include at most *one* application in this domain; the support for more complex services is left to our future work.

not supported by the vanilla OpenStack software, which does not allow a fine control of the network traffic coming from the external world. The Openstack domain orchestrator solves this issue, as shown in Figure 3. Upon receiving a service request, the OpenStack domain orchestrator sends part of the service to the OpenStack controller that takes care of starting the NFs and creating connections among them. It also interacts directly with the SDN controller to set up the inter-domain traffic steering (e.g., create a GRE tunnel linked to the external domain). Finally, the OpenStack domain orchestrator creates the list of available services based on the content of the OpenStack image repository, which allows the Telco orchestrator to recognize whether the considered domain supports a given application.
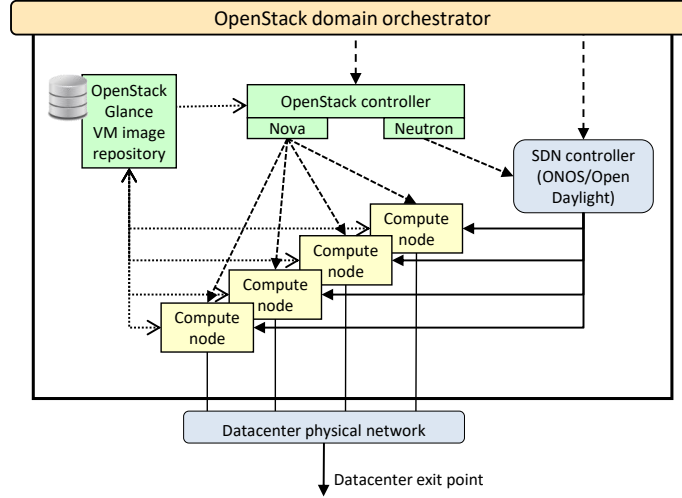

Fig. 3. Logical structure of the OpenStack domain.

4.3 Universal Node domain orchestrator

The Universal Node (UN) [20] orchestrator can be considered as a "data center in a box", and it handles the orchestration of compute and network resources within a single physical node, such as a standalone server or a resource-constrained Customer Premises Equipment. The UN supports multiple types of NFs that can be mixed together when creating a service chain, such as VMs, Docker, DPDK processes, and Native Network Functions (NNF) [21]. Particularly, NNFs exploit the presence of native software modules (e.g., `iptables`) and (often) hardware accelerator (e.g., crypto hardware, integrated L2 switch) that are often available in Linux-based embedded devices. Since, in our model, NFs are exported as *capabilities* to the overarching orchestrator, no matter how the actual implementation is (e.g., VM/Docker/NNF), NNF provides a way to implement NFs with

11

reduced overhead compared to VMs and containers, while still being transparent to the service requester. Thus, a service request can be served by instantiating at least a portion of it at the very edge of the network, on the user residential gateway, while other NFs (e.g., the ones that require more resources) can be allocated in data centers, hence enabling the users to experience a better service given its proximity to the requested NFs.

5 Validation

The prototype of our 5G-OP has been validated on the experimental JOL-NET [22] facility. JOLNET is an Italian geographical testbed consisting of a set of nodes that represent the access side (e.g., residential gateways) and a set of OpenStack domains, connected through a pure OpenFlow network; the Internet can be emulated by installing "public" services on one of the JOLNET POPs, as shown in Figure 4.

The service chain deployed in the testbed is shown in Figure 4(a). It operates between the user U at the edge of the network and the server S on the Internet and includes two NFs, a firewall and a NAT that can have different implementations in each domain. Indeed, the firewall is a NNF exploiting a simple `iptables` executed on the UN, while the NAT can be either an ONOS bundle running in SDN domain or a KVM-based VM in the data center. The specific domain where to start each NF is up to the Telco orchestrator, whose current logic aims at allocating NFs as close as possible to the traffic source (user U), subject to the availability of that NF in the given domain, according to the information exported by each domain controller and shown with *(1)* in Figure 4.

In the first validation scenario, shown in Figure 4(a), the firewall is deployed in the UN because this domain represents the entry point of the user's traffic and the domain orchestrator exports this capability. Instead, the NAT is deployed in the data center in Venice, as this is the only domain advertising such a capability; the SDN network is used only to create the logical connection (through the proper VLAN ID) between the two portions of the service. In the second validation scenario, shown in Figure 4(b), the NAT capability is exported also by SDN domain orchestrator, hence the Telco orchestrator selects this domain for the execution of this NF, thus reducing the distance between chained NFs and freeing up all the consumed resources in the OpenStack domain.

In both cases, we measured the end-to-end latency (using the `ping` command) and the throughput (generating TCP traffic through `iperf`) between the user U and the server S.

As shown in Table 1, performance results improve when the NAT is executed in the SDN domain (case (b)), with respect to the case in which such a NF is deployed in the data center (case (a)). It happens because *(i)* the traffic does not need to be forwarded to the distant data center before being actually delivered to S, and *(ii)* the NAT is actually implemented as a set of Openflow rules installed by the ONOS NAT bundle in the hardware switches (only the first packet of a flow is processed in the software bundle, while the following packets are directly
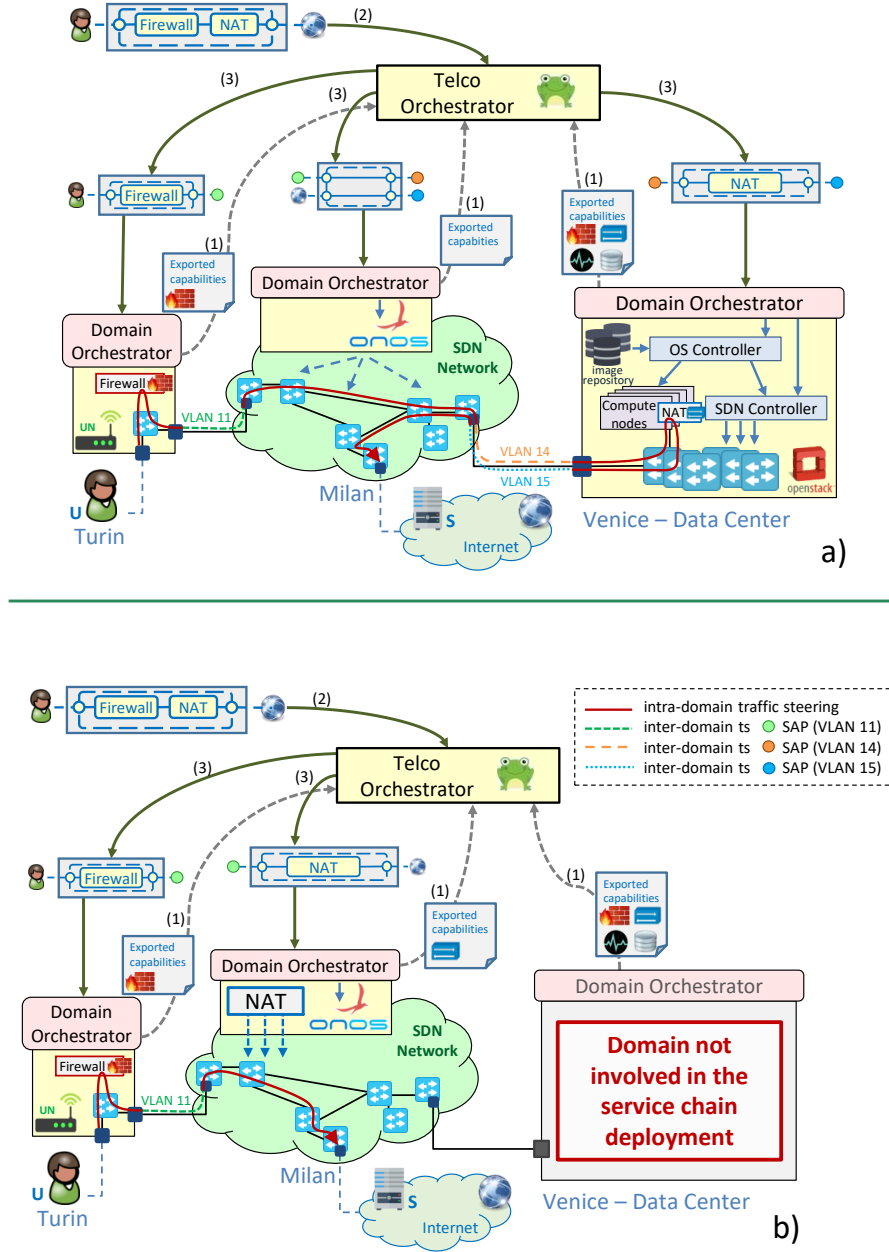
Fig. 4. Validation scenario: (a) service chain split across three domains; (b) service chain deployed in two domains.

Table 1. Comparing the throughput when different domains are involved in the service deployment.

Test case	TCP throughput	End-to-end latency
Baseline: direct connection to Internet, no NFs	88.59 Mbit/s	10.96 ms
Case (a): NFs spanning across three domains	67.04 Mbit/s	27.10 ms
Case (b): NFs spanning across two domains	81.34 Mbit/s	14.99 ms

processed in hardware by the switches). Notably, the result achieved in case (*b*) are very close to the case in which no network function is present (*baseline*). In all cases, the throughput is limited by the speed of the geographical connections (100Mbps).

An early version of the prototype has been showcased at the ITU IMT2020 Workshop and Demo Day [23].

6   Conclusions

This paper presents the reference functional architecture of a unifying Operating Platform for 5G (5G-OP) integrating management, control and orchestration of computing, storage and networking resources down to the end-user devices and terminals (e.g., smart phone, machines, robots, drones, autonomous vehicles, etc). 5G-OP represents an attempt to overcome the fragmentation in Standardization Bodies and large-scale open source development communities, where it is still difficult to find consolidated control and orchestration solutions that can be easily taken up by Telco operators and service providers. The agnostic and overarching characteristics of the 5G-OP will allow decoupling from the underneath control-orchestration platforms, which thus become pluggable in the proposed platform.

Future work on 5G-OP is planned to evolve the FROGv4 open-source prototype and add more 5G-OP functions across more technological network and non-network domains. The enhanced prototype will be validated against some of the most challenging 5G use cases briefly introduced in this paper, thus providing a solid reference framework for Telcos to efficiently operate the 5G networks.

References

1. A. Manzalini, R. Minerva, F. Callegati, W. Cerroni, and A. Campi, "Clouds of virtual machines in edge networks," *IEEE Communications Magazine*, vol. 51, no. 7, pp. 63–70, 2013.
2. R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, "Network function virtualization: State-of-the-art and research challenges," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 236–262, 2016.
3. D. Kreutz *et al.*, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, 2015.

4. A. Checko, H. L. Christiansen, Y. Yan, L. Scolari, G. Kardaras, M. S. Berger, and L. Dittmann, "Cloud RAN for mobile networks - A technology overview," *IEEE Communications surveys & tutorials*, vol. 17, no. 1, pp. 405–426, 2015.

5. "The OpenDayLight open source project." http://www.opendaylight.org/.

6. P. Berde *et al.*, "ONOS: towards an open, distributed SDN OS," in *Proceedings of the third workshop on Hot topics in software defined networking*, pp. 1–6, ACM, 2014.

7. "The OpenStack project." http://www.openstack.org/.

8. M. Di Mauro, F. Postiglione, M. Longo, R. Restaino, and M. Tambasco, "Availability evaluation of the virtualized infrastructure manager in Network Function Virtualization environments," in *Risk, Reliability and Safety: Innovating Theory and Practice*, pp. 2591–2596, Walls, L. and Revie, M. and Bedford, T., 2016.

9. D. Kakadia, *Apache Mesos Essentials*. Packt Publishing, 2015.

10. "The OpenSouce MANO project." https://osm.etsi.org/.

11. L. Peterson *et al.*, "Central office re-architected as a data center," *IEEE Communications Magazine*, vol. 54, no. 10, pp. 96–101, 2016.

12. M. Yoon, T. Tofigh, and G. Parulkar, "Rethinking the Mobile Edge Network with CORD (Mobile-CORD)," *IEEE SDN Newsletter*, March 2016.

13. "The Robot Operating System (ROS)." http://wiki.ros.org/ROS/Introduction.

14. F. Paganelli, M. Ulema, and B. Martini, "Context-aware service composition and delivery in NGSONs over SDN," *IEEE Communications Magazine*, vol. 52, no. 8, pp. 97–105, 2014.

15. L. Csikor, M. Szalay, B. Sonkoly, and L. Toka, "Nfpa: Network function performance analyzer," in *IEEE Conference on Network Function Virtualization and Software Defined Network (NFV-SDN)*, pp. 15–17, IEEE, 2015.

16. "IETF Benchmarking Methodology WG." https://tools.ietf.org/wg/bmwg/.

17. "The FROG open source project." https://github.com/netgroup-polito/frog4.

18. "The DoubleDecker project." http://acreo.github.io/DoubleDecker/.

19. M. Bjorklund, "The YANG 1.1 Data Modeling Language." RFC 7950, Aug. 2016.

20. I. Cerrato, A. Palesandro, F. Risso, M. Suñé, V. Vercellone, and H. Woesner, "Toward dynamic virtualized network services in telecom operator networks," *Computer Networks*, vol. 92, pp. 380–395, 2015.

21. R. Bonafiglia, S. Miano, S. Nuccio, F. Risso, and A. Sapio, "Enabling NFV Services on Resource-Constrained CPEs," in *5th IEEE International Conference on Cloud Networking (Cloudnet)*, pp. 83–88, IEEE, 2016.

22. "JOLNET: a geographical SDN network testbed." https://www.softfire.eu/jolnet/. Accessed on: 2017-02-06.

23. A. Manzalini, G. Castellano, and F. Risso, "5G Operating Platform: Infrastructure-agnostic orchestration." in *IMT2020 Workshop and Demo Day: Technology Enablers for 5G*. ITU, December 2016. Youtube video recording available https://www.youtube.com/watch?v=N6SBo2f6Lyc.