2017 IEEE 31st International Conference on Advanced Information Networking and Applications

Maestro: An NFV Orchestrator for Wireless Environments Aware of VNF Internal Compositions

Ariel Galante Dalla-Costa*, Lucas Bondan*, Juliano Araujo Wickboldt*,
Cristiano Bonato Both†, Lisandro Zambenedetti Granville*
*Institute of Informatics – Federal University of Rio Grande do Sul
†Department of Applied Mathematical and Social Sciences – Federal University of Health Sciences of Porto Alegre
Email: {agdcosta, lbondan, jwickboldt, granville}@inf.ufrgs.br, cbboth@ufcspa.edu.br

***Abstract*—Dynamic Cloud Radio Access Network (Dynamic C-RAN) is an emerging wireless architecture that aims for flexibility, business agility, adaptability, among other benefits. In a Dynamic C-RAN, wireless functionalities can be split into smaller components and distributed along a hierarchical cloud infrastructure. Network Functions Virtualization (NFV) concepts have been recently investigated to facilitate management-related operations of these wireless functionalities. Despite the many advocated advantages of the function's splitting and the effectiveness of NFV orchestration solutions, both academia and industry are considering Virtualized Network Functions (VNF)s as atomic elements, disregarding the potential advantages of splitting VNFs into several different components. Aiming to improve VNF orchestration in Dynamic C-RAN scenarios, in this paper we propose *Maestro*: an NFV orchestrator for wireless environments that is able to decide among several possible VNF compositions which are more suitable for each situation. *Maestro* is designed to operate using different decision mechanisms that can be defined based on network operators' needs. We evaluate the effectiveness of our proposal by modeling the orchestrator's decision mechanism as a linear programming problem. Thus, we show how fronthaul bandwidth consumption can be reduced threefold considering different VNF compositions against atomic VNF placement.**

***Index Terms*—Dynamic C-RAN, NFV orchestration, VNF composition**

I. Introduction

Dynamic Cloud Radio Access Networks (Dynamic C-RAN) move functions from the radio front-end to the network core in a cloud-based infrastructure, providing more flexibility for service provisioning [1]. The dynamic deployment of radio functions in Dynamic C-RAN reduces Capital and Operational Expenditures (CAPEX and OPEX), at the same time it turns the management of such functions easier than the management of traditional dedicated devices. In Dynamic C-RAN, Base Station (BS) processes may be split according to their functionalities and placed in different locations of the infrastructure, which results in benefits and opportunities, such as adaptability, load balancing, service deployment flexibility, and energy efficiency [2].

Despite the benefits of Dynamic C-RAN, coordinating radio functionalities operation is not an easy task, since the path along these functions must support strict requirements, such as low delay and minimum data rate [3] [4]. For this reason, both academia and industry have been exploiting the employment of Network Functions Virtualization (NFV) concepts to manage and orchestrate virtualized radio functionalities in next generation wireless networks [5]. The NFV architecture allows network operators to design real-time automated orchestration solutions, easing operations such as network functions monitoring, upgrading, and maintenance, when compared to traditional network functions management [6].

NFV platforms such as Cloud4NFV [7] and T-NOVA [8] employ NFV Orchestrators (NFVO) for Virtualized Network Functions (VNF) provisioning. These platforms consider VNFs as atomic elements, *i.e.,* indivisible functions. Despite useful, this approach does not take full advantage of the NFV architecture, whereas VNFs can be presented in different compositions split in different Virtual Deployment Units (VDU), enabling the creation of VNFs that perform the same function, but with different internal composition. For example, a VNF of a Baseband Unit (BBU) may be placed in two different ways: totally deployed in the radio front-end or spread along a Dynamic C-RAN infrastructure, depending on its composition.

Taking into account the lack of NFVO aware of VNF internal compositions, we propose *Maestro*: an NFVO for wireless networks focused on the decision process regarding VNF deployment. *Maestro* is designed to adapt VNF operation based on a catalog with different VNF implementations. This catalog consists of VNFs that perform the same network function but are represented by different VDUs. VNFs with different compositions lead to a decision problem, were the NFVO must decide which is the best VNF composition to deploy, considering the resources available on the Dynamic C-RAN and monitored information regarding VNFs operation.

We advance the state of this research topic through the following contributions: (*i*) proposing *Maestro*, an NFV orchestrator aware of internal VNF compositions, (*ii*) detailing the NFV management and orchestration definitions found in the literature and applying these definitions to Dynamic C-RAN, and (*iii*) presenting a case study and modeling the orchestrator's decision mechanism as a linear programming problem showing *Maestro*'s advantages in comparison with standard NFV orchestration in a Dynamic C-RAN environment. The results obtained show that, when comparing the use of atomic VNFs and VNFs with multiple VDUs, there is a significant decrease of data rate consumed in the fronthaul network in the order of 70% and, on the other hand, an increase in the VDU placement calculation time.

1550-445X/17 $31.00 © 2017 IEEE
DOI 10.1109/AINA.2017.126

The remaining of this paper is organized as follows. In Section II, we present the background and related work on Dynamic C-RAN and NFV Management and Orchestration (MANO). In Section III we present *Maestro*, our solution for NFV orchestration in wireless environments. Based on the proposed solution, a case study and an evaluation model based on linear programming are presented in Section IV. In Section V we present and discuss the results obtained during regarding the proposed evaluation model. Finally, in Section VI we present final remarks and perspectives for future work.

II. Background and Related Work

In this section, we provide a background on Dynamic C-RAN, that is the targeted scenario of our proposal. Next, we discuss important works in this area and motivate our research by discussing the applicability of NFV in Dynamic C-RAN environments.

A. Dynamic C-RAN

Dynamic C-RAN is a dynamic implementation of Cloud Radio Access Networks (C-RAN) that is a centralized architecture for wireless networks [9]. Different from C-RAN, where all radio-related functions are moved to the cloud, in Dynamic C-RANs baseband functions are usually kept close to the BSs, while higher level functions are deployed in distributed clouds. This distribution has many advantages when compared to C-RAN, such as energy efficiency, cost saving on Capital and Operational Expenses (CAPEX and OPEX), capacity improvement, more flexibility due to the dynamic deployment of functions, and service adaptability. For this reason, 5th Generation (5G) Mobile Networks foresees operating in Dynamic C-RANs [1].

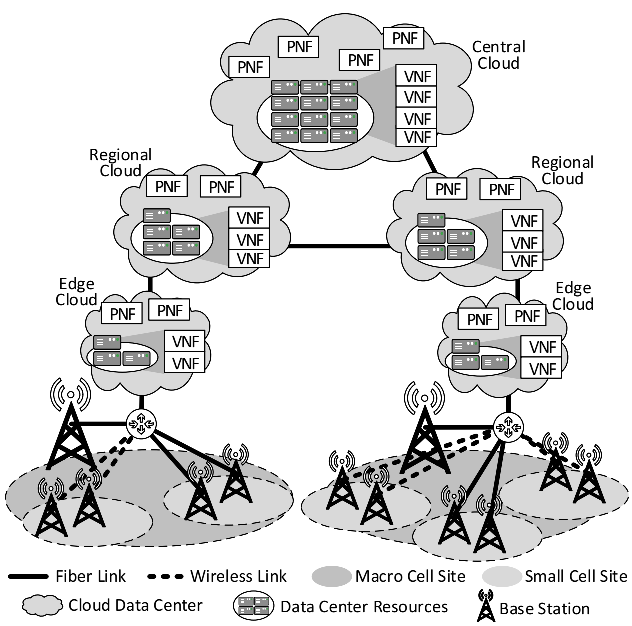

Fig. 1. Example of Dynamic C-RAN scenario

Our Dynamic C-RAN scenario is composed of cloud data centers hierarchically organized, similar to the geographical distribution of clouds in Fog computing scenarios [10], as depicted in Fig. 1. These clouds are composed of both legacy Physical Network Functions (PNF) and generic services for hosting VNFs. The Central Cloud is usually at the operator's central office and provides the highest resource volume in the infrastructure. Regional and Edge clouds are closer to the radio front-end, usually providing fewer resources than the Central Cloud. The hierarchical strategy enables the dynamic placement of virtualized functions, turning service provisioning more flexible [11]. For example, a new virtualized Baseband Unit (BBU) can be deployed on a Regional Cloud to accommodate the increasing number of users in a specific Edge Cloud. Moreover, enforcing load balancing and energy saving policies become easier, due to the flexibility for migrating VNFs across data centers.

Coordinating Dynamic C-RAN with virtualized BBU functions is not an easy task. The proper division of these functions is an important research problem currently explored in this area. In their work, Abdelwahab *et al.*[3] and Liu *et al.*[4] present splitting proposals and strategies for BBU functions. These works recognize that NFV can play an important role in the management and orchestration of virtualized wireless functions [5], [12]. However, using NFV in this kind of network leads to another research problem regarding how to orchestrate virtualized wireless functions across the Dynamic C-RAN infrastructure properly.

In this paper, we abstract the problem of dividing BBU functions, assuming that any validated division can be considered in the orchestration process, such as the proposals of Abdelwahab *et al.* [3] and Liu *et al.* [4]. Our focus resides on the orchestration of VNFs representing BBU functions, considering different compositions for the same VNF. In the next subsection, we provide a detailed explanation of the NFV architecture, focusing on the functional blocks responsible for MANO operations, presenting some works found in the literature regarding NFV orchestration, and discussing their applicability in Dynamic C-RAN environments.

B. NFV Management and Orchestration

Initiated by the world's leading telecommunications network operators, the NFV Industry Specification Group (ISG) created under the European Telecommunications Standards Institute (ETSI) coordination is working on a consensus regarding the virtualization of network functions, focusing on providing integration among NFV solutions [13]. Initially, the NFV ISG created an architecture where the central elements are the VNFs, with complementary functional blocks involved in their operation. As the NFV concept evolved, the management and orchestration of VNFs became an important issue for the proper operation of NFV environments, leading the NFV ISG to create the NFV MANO group, focused on specifying the operation of these mechanisms [14]. In Fig. 2 we present the NFV architecture proposed by ETSI, detailing the functional blocks regarding NFV MANO.

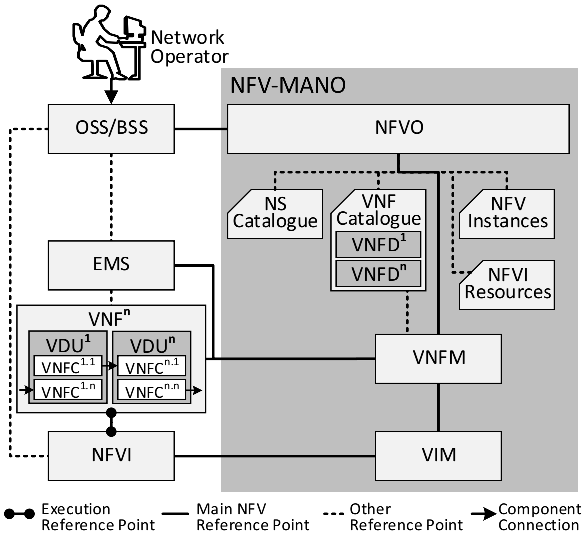

Fig. 2. NFV architecture with detailed MANO blocks

The central elements of the NFV architecture are the VNFs that are software implementations of NFs deployed in the Network Function Virtualization Infrastructure (NFVI). A VNF is composed of VDUs, which are the smallest elements of a VNF, directly mapped on a dedicated virtualization container (*e.g.,* virtual machine, Linux container) [14]. Likewise, each VNF is described by a VNF Descriptor (VNFD) that details how VDUs are connected to compose the VNF. Each instance of a deployed VDU is represented by a VNF Component (VNFC), which also handles the connections points between VDU instances. In other words, a VNF could be represented by different VDUs (*i.e.,* the image of a VM/container for each VNF composition), while each instance of a given VDU is represented by a VNFC (*i.e.,* the VM/container deployed). All these elements must be considered by NFV orchestration solutions designed to operate with VNFs presenting different internal compositions.

All functional blocks of the NFV architecture are connected to NFV MANO elements, responsible for managing and orchestrating services, functions, and resources. Services and functions should be registered by network operators, respecting access control rules enforced by Operations/Business Support Systems (OSS/BSS). Information regarding VNF composition, instantiation, and operation are handled by the NFV Orchestrator (NFVO), which is the NFV element responsible for bringing intelligence to service provisioning and composition processes. For this reason, the NFVO is responsible for two main tasks: (*i*) resource orchestration across multiple Virtual Infrastructure Managers (VIM), fulfilling resource orchestration requirements, and (*ii*) life-cycle management of network services composed by VNFs [14], interacting directly with different VNF Managers (VNFM).

NFVO has access to the Network Services (NS) and VNF Catalogs, that maintain information regarding available services and functions. Furthermore, the NFVO also has access to NFV instances operating in the NFVI, and to physical and virtual resources of the NFVI. All this information can be used by the NFVO to take decisions about network services and functions operation, performing actions such as scale up or down VNF resources and migrating VNFs across the NFVI. For this reason, the orchestration of NFV services and VNFs is a key process for the proper operation of NFV-based environments, since it has a direct impact on the network performance [15].

Taking into account the important role played by the NFVO in the NFV architecture, some proposals have emerged in the literature. For example, OpenMANO is an open-source based on an implementation of ETSI NFV MANO framework focused on performance and portability [5]. Moreover, NFV platforms, such as Cloud4NFV [7] and T-NOVA [8], propose NFVOs for VNF provisioning focused on end-to-end services. Munoz *et al.* [16] propose an orchestrator for hybrid Software-Defined Networking (SDN) fiber data center environments based on the migration of virtual SDN controllers across heterogeneous infrastructures. All solutions cited above are designed to operate on any network, but they do not provide any directions for the orchestration of wireless-based virtualized functions.

The splitting of wireless functions can substantial important gains for wireless networks. Proposals like Abdelwahab *et al.* [3] and Liu *et al.* [4] highlight the advantages and challenges in splitting BBU's functions, but the orchestration of VNFs composed by different elements remains an open question. At the best of our knowledge, *Maestro* is the first NFV orchestrator able to handle VNFs considering this granularity. In the next section, we present in details the functional blocks of *Maestro*, explaining its operation.

III. NFV Orchestrator

Maestro was designed to fulfill the lack of NFVOs aware of VNF composition, following the recommendations of ETSI's NFV MANO framework [14]. For this reason, *Maestro* can operate in any NFV environment that follows ETSI's architecture. Fig. 3 depicts in details all functional blocks that compose *Maestro* architecture and their relationship with the NFV functional blocks defined by ETSI.

In an NFV environment, network operators are responsible for selecting and configuring all network services and functions, respecting access control rules enforced by OSS/BSS. *Maestro* communicates with OSS/BSS, enabling network operators to configure services, functions, and the NFVI. Moreover, some functional blocks of *Maestro* can also be configured according to network operators needs. All functional blocks designed in *Maestro* operation are described as follows.

Register: According to NFV MANO framework, an NFVO has access to different catalogs regarding available network services and functions, as the instances currently running and resources available in the NFVI. This information may be used by the NFVO to perform actions regarding the operation of

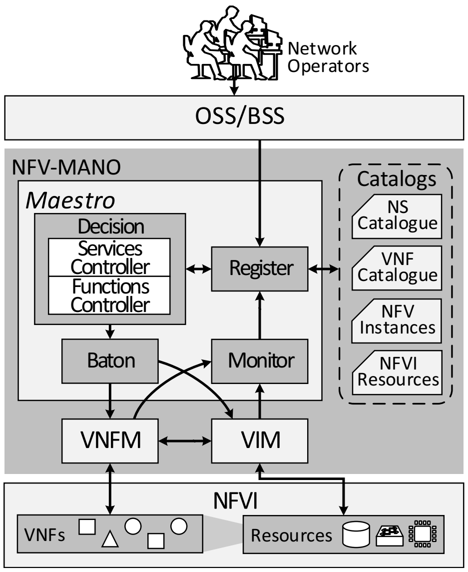

Fig. 3. *Maestro* Architecture

services and functions. In *Maestro* architecture, this information are handled by the Register block. Any new service or VNF must be registered in the catalogs to be available for instantiation in the NFVI.

Catalogs: Store data regarding available services, functions, and resources in the network, as the current status of the instances operating on it, following the information model proposed by ETSI NFV MANO [14]. All catalogs are connected to the Register block, which controls the access to this data by the other functional blocks.

Decision: Determine which actions will be performed in the network, such as scaling up/down VNF's resources, instantiate a new VNF, or change a VNF composition for another one. The Decision block is the central element of *Maestro* architecture, capable of operating with different decision mechanisms, which could be defined before starting *Maestro*'s operation. Each mechanism can consider different parameters to take its decision, such as data rate requirements, information about physical resources (*e.g.,* CPU and memory usage), latency, delay, response time, and network throughput. Regardless the decision mechanism used, the Decision block uses information residing in the catalogs to decide the actions to be taken. These actions can affect the network at two different levels: service or function. For this reason, the Decision block is subdivided into two modules:

- *Services Controller*: Responsible for managing the Service Functions Chainings (SFC), represented as VNF Forwarding Graphs (VNFFG) in the NFV architecture [17], [13]. Once registered in the catalog, an SFC can be instantiated and modified by the decision mechanism implemented in *Maestro*. This module decides the operation regarding the SFC life-cycle, such as instantiation, validation, authorization, integrity administration, and visibility. Moreover, in an NFV environment ruled by OSS/BSS policies regarding the SFC operation, this module is responsible for guarantee the enforcement of these policies;
- *Functions Controller*: Responsible for handling VNF-related operations and taking the decisions based on the available information and the decision mechanism implemented. Once registered in the catalog, a VNF will be considered in the decision process, either for deploying a new service or changing a VNF already running in the NFVI for a new one with a different composition.

Baton: Once defined the actions to be performed regarding services, VNFs, or resources, they must be sent to the respective blocks. The Baton block is responsible for receiving these actions from the Decision block, translating it into configurations, and forwarding it to the VIM and/or VNFM. These two functional blocks will apply the configurations sent by the Baton block according to their implementation.

Monitor: Monitors and collects data regarding VNFs operation and NFVI resources, forwarding all data collected to the Register block. The Monitor block operates in two ways: (*i*) oriented by events, such as VNF or VM malfunctioning; and (*ii*) monitoring by polling, performing periodical requests to VNFMs and VIM attempting information such as available resources, data rate consumption, and network latency. Event-related information is generated by VNFM and VIM and does not need the intervention of the Monitor block, while the Monitor periodically requests data in the polling mode.

Both Baton and Monitoring blocks are connected to VIM and VNFM blocks from the ETSI NFV architecture. These two blocks are responsible for handle VNF operation and the resources of the NFVI, respectively. *Maestro* was designed to operate with any implementation of VIM and VNFM that follow the ETSI's recommendations for NFV implementations, such as Cisco NSO [18], OpenStack [19], and Aurora [20]. Changing from one solution to another requires only adapting the abstraction layer present in both Baton and Monitoring blocks of *Maestro*. Once introduced *Maestro* architecture, in the next section we describe a case study and the evaluation modeling designed as a proof-of-concept of *Maestro* operation.

IV. Case Study and Evaluation Model

We defined a case study to demonstrate the efficiency of *Maestro* in selecting an appropriate VNF deployment. Based on this case study, we modeled *Maestro* operation to evaluate the resulting set of VDUs selected and their efficiency in fulfilling a set of predefined requirements. Both the case study and the evaluation modeling are described in details in the next subsections.

A. Case Study

Our case study is based on wireless radio functions split presented by Wubben *et al.* [21], applied to the context of Dynamic C-RAN. This way of splitting is interesting in this context due to the variability of the fronthaul data rate demand imposed by each split option, being divided into 5 functions:

MAC, Soft-Bit, RX Data, Subframe, and I/Q. More details about the operation of each function can be found in the work of Wubben *et al.* [21]. This case study allows us to demonstrate the use of NFV MANO elements (VNFs, VDUs, and VNFCs) in the orchestration of Dynamic C-RAN and how different splits may impact in the network performance and resources usage. For this reason, the splits used are based on the communication and computing requirements of the functions that compose the wireless radio functions. Fig. 4 depicts our case study, showing benefits of the split deployment against the atomic split, as proposed by Wubben *et al.* [21]. Besides proposing the splits, the authors defined communication requirements among each of the functions that must be considered by the NFVO in the decision process.

The BSs are depicted at the bottom of Fig. 4, each one with a different occupation. The macro cell (b) presents an occupation of 90%, while small cell (a and c) are 10% and 20% occupied, respectively. The workload of the BSs is distributed into three cloud levels: (*i*) Edge, close to the BSs; (*ii*) Regional, a little bit far from the BSs; and (*iii*) Central, even farther from the BSs, usually on the service provider premises. Each cloud level has a different processing capacity: the farther from the BSs, the greater the computing capacity of it. In our case study, the computational capacity of the cloud is measured regarding of computing unities. The Edge cloud provides 5 computing unities, while Regional and Central clouds provide 7 and 15 computing unities, respectively. Since our objective with this case study is analyzing the fulfillment of network requirements, we simplified the computational analysis considering that each virtualized component (*i.e.,* VNFC) consumes one computing unity. It means that the most computing intensive VNFCs will fill an entire computing unit, whereas less intensive VNFCs will also have one dedicated computing unity, that will be underused.

Assuming as an example that the objective is set for saving fronthaul data rate, the NFVO cannot improve further the allocation shown at the left side of Fig. 4, since the 5 VNFCs must be moved and allocated together based on the same VDU. In this deployment, the aggregated data rate consumption is 4.8 Gbps between Edge and Regional clouds. The atomic deployment is also suboptimal in this case because it creates a fragmented situation regarding computing unities in use, where only 5 of the 7 computing unities available in the Regional cloud are allocated (*i.e.,* 71% of computing resources usage). With no more options to split VNFs, *i.e.,*VDUs, there is no VNFC able to use the remaining 2 computing unities. In the Split-aware deployment, however, the allocations drastically change. For optimizing data rate in the fronthaul, the NFVO can prioritize combinations of splits with smaller aggregated data rate consumption to be allocated, considering the maximal computational capacity of the Edge and Regional clouds. An advantage of this approach is the usage of all computational resources on the Regional cloud, which was not possible in the Atomic deployment due to the indivisibility of the components of the same VNF. The fronthaul data rate consumption optimization is an example of NFVO objective. Using *Maestro*, network operators can define decision algorithms with different goals.

In the case study presented we consider the BS occupation influences the data rate consumption requirement of functions from RX Data and above, as described by Wubben *et al.* [21]. As a consequence, VDUs containing Subframe and I/Q functions were deployed closer to the BSs, since their data rate requirements are higher. For example, the NFVO could select any function to be deployed at the Central cloud. However, allocating the functions of BS b in the Central cloud will consume much more data rate, since the BS b presents higher occupation than the others, so the data rate requirement is also higher than a and c.

In summary, the case study discussed in this subsection allows us to observe two main advantages of the split-aware deployment compared to the atomic deployment: (*i*) computing resource fragmentation is reduced because of the flexible allocation of different VDU compositions and (*ii*) the aggregated fronthaul data rate consumption is reduced considering the specific requirements of each split even though the same amount of signal processing workload (same BS occupation) was used in both deployment scenarios. Based on this advantages, we modeled an evaluation as a linear programming problem, explained in details in the next subsection.

B. Evaluation Model

To evaluate *Maestro* architecture, we defined a decision mechanism based on the case study presented in the previous subsections. The decision mechanism was modeled as a linear programming problem, based on the solution proposed by Luizelli *et al.* [22] for the VNF placement problem. Our decision solution, however, considers computing capacity and fronthaul data rate requirements for choosing the best VDU combination in different datacenters, while Luizelli *et al.* only considered the placement of atomic VNFs in one datacenter. The parameters used in our model are presented in Table I.

TABLE I
EVALUATION PARAMETERS

Parameter	Description
V	VDU list
H	Host list
A	Base Station list
DB	Fronthaul data rate cost
DV	VDU computational cost
CH	Host computational capacity
CB	Host fronthaul data rate capacity
x	Tuple (v, h, a), where $\forall v, h, a \in (V, H, A)$
ω	Data rate decision weight

Our model receives as input the lists of VDUs, hosts, and BSs (V, H, and A, respectively). For the sake of simplicity, in this experiment we consider one host with a given capacity to represent a whole datacenter, *i.e.,* we disregard the internal organization or topology of a datacenter when calculating fronthaul data rate requirements. DB is the cost of the VNFC traffic to be transported from the BS to the place where it will be allocated. Similarly, DV represents the amount of

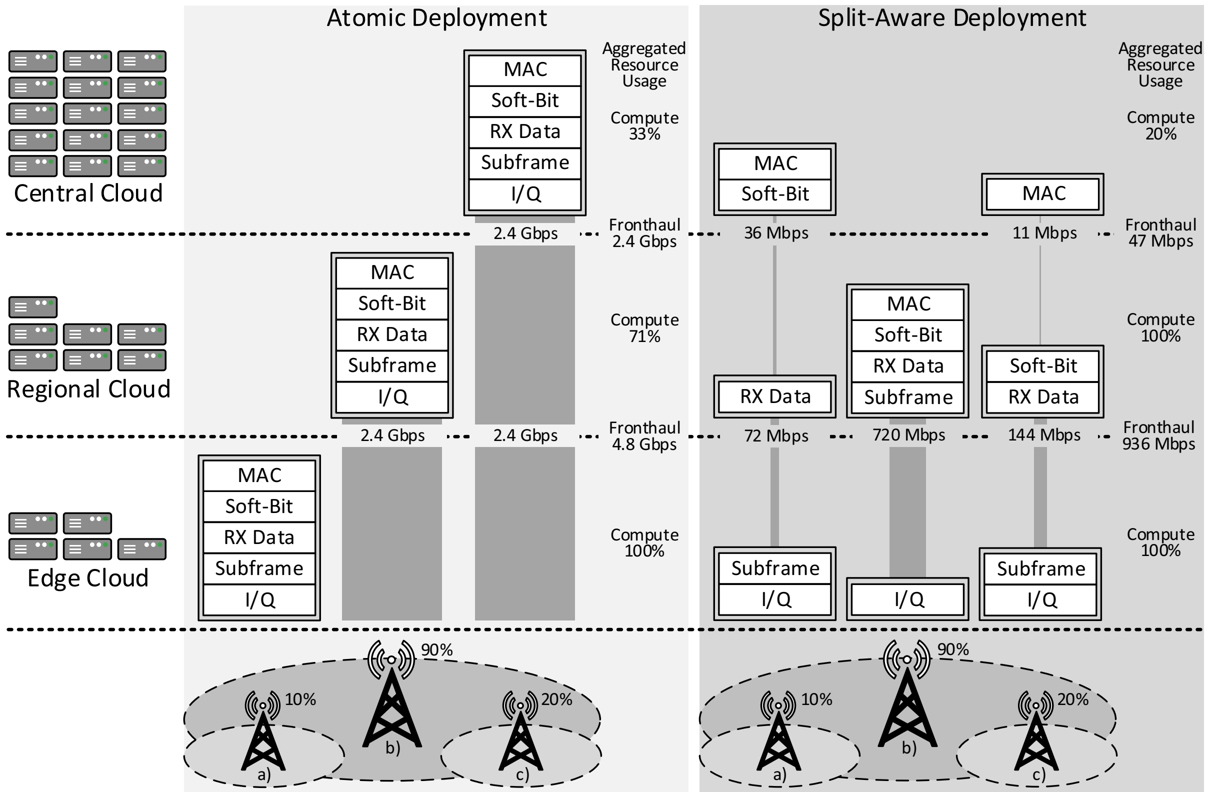

Fig. 4. Atomic and split-aware orchestration examples

computing resources the VDU requires from the hosts. CH represents the number of computing unities of a given hots. In the same way, CB is the maximum data rate that can be consumed in a given host. The combination of VNFC, host, and BS is represented by x. Finally, we defined a weight for the decision process regarding the importance of the data rate (ω). Once set the parameters of our evaluation, we designed the fitness function of evaluation model as follows.

$$min\left(\omega\sum_{v=0}^{V}\sum_{a=0}^{A}\sum_{h=0}^{H}x(v,h,a)DB(v)h+ +(\omega-1)\sum_{v=0}^{V}\sum_{a=0}^{A}x(v,0,a)DV(v)\right) \tag{1}$$

The objective is minimizing the fronthaul data rate consumption, maximizing the resource usage in the clouds close to the BSs (Edge and Regional). The fitness function must obey two main restrictions: (*i*) not exceeding the computational capacity of the hosts and (*ii*) not exceeding the fronthaul capacity. These two restrictions are represented by Equations 2 and 3, respectively.

$$\forall h\in H,\sum_{a=0}^{A}\sum_{v=0}^{V}x(v,h,a)DV(v)\leq CH(h) \tag{2}$$

$$\forall h\in H,\sum_{v=0}^{V}\sum_{a=0}^{A}x(v,h,a)DB(v)\leq CB(h) \tag{3}$$

Once defined the parameters and the model to be evaluated, in the next, we present and discuss the results obtained during the evaluation of the proposed model.

V. Results & Discussion

We have implemented the model using IBM's CPLEX Optimizer [23]. The simulation was run on a computer with Intel Core i7 processor 2.4 GHz with 8 GB RAM, running a Linux operating system. We decided to vary the number of antennas in 9 scenarios between 10 and 1000 to understand the impact it would pose to both the fronthaul data rate consumption and the time to calculate VNFC placement. For all scenarios, we considered three datacenters (*i.e.,* Edge, Regional, and Central) with just combined capacity to accommodate the signal processing workload of the antennas proportional to the examples in Fig. 4 and antenna occupation fixed in 50%. All scenarios have been repeated 20 times since the time measured by the CPLEX software to calculate VNFC placement varies in each execution.

Fig. 5 shows the aggregate data rate consumption in the fronthaul network for different forms of VNF split. The x-axis indicates the number of antennas in each experimentation scenario and the y-axis (log-scaled) shows the amount of aggregate fronthaul data rate consumed in Gbps given the placement of VNFCs calculated by the solver. The *Atomic split* line represents the data rate consumed when only one VDU is used to deploy each VNF, *i.e.,* each antenna will have its five VNFCs positioned in the same cloud. The line *1 and 2 VDU split* shows data rate consumption in the situation in which the solver can use single VDU deployments (like the *Atomic split* line) as well as two VDUs per VNF. Finally, the *1, 2, and 3 VDU split* line show the amount of data rate consumed in the fronthaul by deployments up to 3 VDUs per VNF (similar to the examples in Fig. 4 *Split-Aware Deployment*).

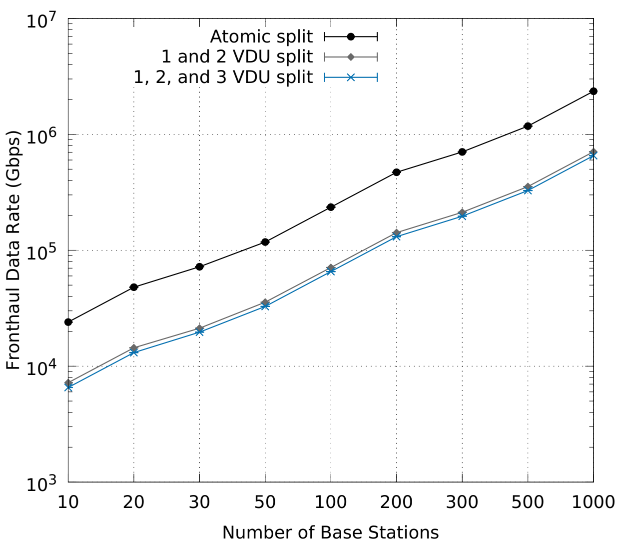

Fig. 5. Aggregate fronthaul data rate consumption

Analyzing Fig. 5, it is possible to visualize that there is a major reduction in data rate consumption when comparing the *Atomic split* line with any of the other two lines for any experimentation scenario. This reduction represents a clear advantage of the *Split-Aware Deployment*, which accounts for a consistent threefold reduction of fronthaul consumption as the amount of VDUs per VNF increases. For example, in the 1000 antennas scenario, the atomic split would consume up to 2.35 Tbps, whereas using 2 and 3 VDUs per VNF these numbers decrease to 708.12 Gbps and 655.2 Gbps, respectively. The gain obtained is not so beneficial when comparing *1 and 2 VDU split* and *1, 2, and 3 VDU split* lines. This small reduction in terms of data rate consumption is possible due to the distribution of VNFCs maximizing the processing in the Edge cloud. The increase in the number VDUs available for deployment is useful to save fronthaul data rate, but also introduces complexity in the VNFCs' placement process as we discuss next.

Fig. 6 illustrates the average time elapsed to compute the placement of each VNFC in all available clouds. The time measured by CPLEX varies in each execution, so error bars are shown in the plot representing a confidence interval of 99% for each average time measure. The x-axis shows the number of antennas in each experimentation scenario, while the y-axis (log-scaled) represents the time in seconds required to calculate the placements of VNFCs. As in Fig. 5, the lines represent the atomic VNFs, up to 2 VDU, and up to 3 VDUs split options.

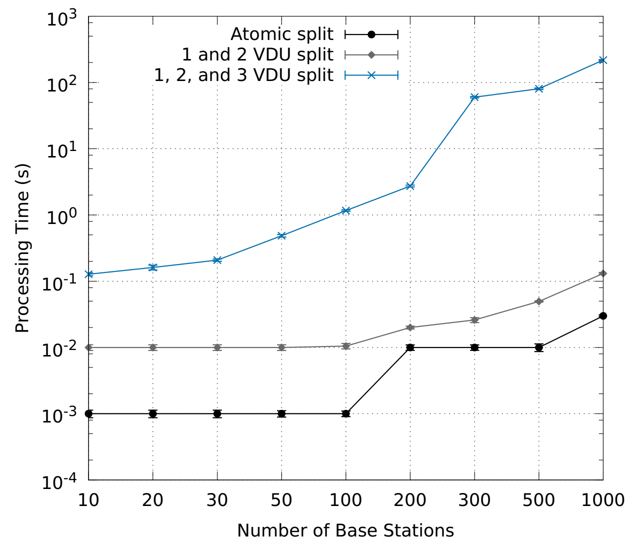

Fig. 6. Time to calculate VNFC placement

From Fig. 6 it is possible to visualize that, as the number of VDUs increases, so does the time for processing the placement of the VNFCs. For the *Atomic split* and *1 and 2 VDU split*, although there is an increase from 100 antennas on, the overall calculation time is almost negligible up to 1000 antennas. The increase in time between *1 and 2 VDU split* and *1, 2, and 3 VDU split*, on the other hand, is substantial. A large increase in processing load can be justified by the number of possible combinations of VDUs that can be used to deploy each VNF.

Comparing both charts of Figs. 5 and 6, it is possible to note that in the former there is gain regarding data rate consumption when considering 3 VDU splits. However, it requires way more processing time for a large number of antennas than 2 VDU splits (217 seconds for 3 VDUs against 0.131 seconds for 2 VDUs for 1000 antennas), which can be an obstacle in scenarios where resource allocations change dynamically.

VI. Conclusion

In this paper, we proposed *Maestro*, an NFV orchestrator for wireless environments aware of VNF internal compositions. The central element of *Maestro* architecture is the decision mechanism, designed for different network operators objectives. Moreover, *Maestro* was designed to operate with any VIM and VNFM implementations following the recommendations of ETSI NFV MANO. As proof-of-concept, we modeled the decision mechanism as a linear programming problem, showing the benefits of VNF composition-aware orchestration.

The results show that *Maestro* has reduced the amount of fronthaul data rate consumed for 1000 antennas from 2.35 Tbps using an atomic split to 708.12 Gbps considering 2 VDUs, and 655.2 Gbps considering 3 VDUs. These results represent a reduction of approximately 69.9% of the data rate consumption for 2 VDUs and 72.1% for 3 VDUs. The reductions of data rate consumption between 2 to 3 VDU splits were in the order of 7.4%. It is also possible to observe an increase of the average time used to calculate the placement of 2 and 3 VNFCs of their respective VDU, increasing from 0.03 seconds in the atomic split to 0.13 seconds in 2 VDUs and 217.2 seconds with 3 VDUs. Through these results, we observe that the improvement in data rate is about the increase in processing time for placement between 2 and 3 VDU splits. This increase in processing time hinders network adaptability facing dynamic resource allocation scenarios.

There are several avenues for future research. First, we plan to improve the experimental modeling by refining the computational capacity analysis considering different needs for each VDU. We also intend to extend *Maestro* operation further by implementing new decision mechanisms based on different network environments. Moreover, *Maestro* can be improved regarding functionalities through new mechanisms to enhance network operation. For example, *Maestro* can consider network policies or indents during the decision process, enforcing network operation according to predefined rules.

Acknowledgment

The research leading to these results received funding from the European Commission H2020 programme under grant agreement no. 688941 (FUTEBOL), as well from the Brazilian Ministry of Science, Technology, Innovation, and Communication (MCTIC) through RNP and CTIC. The authors would also like to thank Anderson Santos da Silva, Maicon Kist, and Marcelo Luizelli for their important contributions during the elaboration and implementation of this work.

References

[1] G. P. W. Group, "View on 5G Architecture," 5G PPP, Tech. Rep., 2016.

[2] J. Bartelt, P. Rost, D. Wubben, J. Lessmann, B. Melis, and G. Fettweis, "Fronthaul and backhaul requirements of flexibly centralized radio access networks," *IEEE Wireless Communications*, vol. 22, no. 5, pp. 105–111, October 2015.

[3] S. Abdelwahab, B. Hamdaoui, M. Guizani, and T. Znati, "Network function virtualization in 5G," *IEEE Communications Magazine*, vol. 54, no. 4, pp. 84–91, April 2016.

[4] J. Liu, S. Zhou, J. Gong, Z. Niu, and S. Xu, "Graph-based framework for flexible baseband function splitting and placement in C-RAN," in *IEEE International Conference on Communications (ICC)*, June 2015, pp. 1958–1963.

[5] R. Mijumbi, J. Serrat, J. l. Gorricho, S. Latre, M. Charalambides, and D. Lopez, "Management and orchestration challenges in network functions virtualization," *IEEE Communications Magazine*, vol. 54, no. 1, pp. 98–105, January 2016.

[6] 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, 2015.

[7] J. Soares, M. Dias, J. Carapinha, B. Parreira, and S. Sargento, "Cloud4NFV: A platform for Virtual Network Functions," in *IEEE International Conference on Cloud Networking (CloudNet)*, Oct 2014, pp. 288–293.

[8] G. Xilouris, E. Trouva, F. Lobillo, J. M. Soares, J. Carapinha, M. J. McGrath, G. Gardikis, P. Paglierani, E. Pallis, L. Zuccaro, Y. Rebahi, and A. Kourtis, "T-nova: A marketplace for virtualized network functions," in *European Conference on Networks and Communications (EuCNC)*, June 2014, pp. 1–5.

[9] K. Chen and R. Duan, "C-RAN The Road Towards Green RAN. White Paper." China Mobile Research Institute, Tech. Rep., 2011.

[10] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog Computing and Its Role in the Internet of Things," in *SIGCOMM Workshop on Mobile Cloud Computing (MCC)*, ser. MCC '12. New York, NY, USA: ACM, 2012, pp. 13–16.

[11] M. Peng, Y. Li, J. Jiang, J. Li, and C. Wang, "Heterogeneous cloud radio access networks: a new perspective for enhancing spectral and energy efficiencies," *IEEE Wireless Communications*, vol. 21, no. 6, pp. 126–135, December 2014.

[12] R. Riggio, T. Rasheed, and R. Narayanan, "Virtual network functions orchestration in enterprise WLANs," in *IFIP/IEEE International Symposium on Integrated Network Management (IM)*, May 2015, pp. 1220–1225.

[13] M. Chiosi *et al.*, "Network Functions Virtualisation (NFV)," ETSI NFV ISG, White Paper 1, 2012, available at: https://portal.etsi.org/NFV/NFV_White_Paper.pdf.

[14] J. Quittek *et al.*, "Network Functions Virtualisation (NFV) - Management and Orchestration," ETSI NFV ISG, White Paper, 2014.

[15] L. Bondan, C. R. P. dos Santos, and L. Z. Granville, "Comparing Virtualization Solutions for NFV Deployment: A Network Management Perspective," in *IEEE Symposium on Computers and Communication (ISCC)*, June 2016, pp. 669–674.

[16] R. Munoz, R. Vilalta, R. Casellas, R. Martinez, T. Szyrkowiec, A. Autenrieth, V. Lopez, and D. Lopez, "Integrated SDN/NFV management and orchestration architecture for dynamic deployment of virtual SDN control instances for virtual tenant networks," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 7, no. 11, pp. B62–B70, November 2015.

[17] P. Quinn and J. Guichard, "Service Function Chaining: Creating a Service Plane via Network Service Headers," *Computer*, vol. 47, no. 11, pp. 38–44, Nov 2014.

[18] Cisco - Network Services Orchestrator Solutions. Available at: http://www.cisco.com/c/en/us/solutions/service-provider/solutions-cloud-providers/network-services-orchestrator-solutions.html. Accessed September, 2016.

[19] R. Bryant *et al.*, "Accelerating NFV Delivery with OpenStack. White Paper," OpenStack Foundation, Tech. Rep., 2016.

[20] J. A. Wickboldt, R. P. Esteves, M. B. de Carvalho, and L. Z. Granville, "Resource management in IaaS cloud platforms made flexible through programmability," *Elsevier Computer Networks*, vol. 68, pp. 54 – 70, 2014.

[21] D. Wubben, P. Rost, J. S. Bartelt, M. Lalam, V. Savin, M. Gorgoglione, A. Dekorsy, and G. Fettweis, "Benefits and Impact of Cloud Computing on 5G Signal Processing: Flexible centralization through cloud-RAN," *IEEE Signal Processing Magazine*, vol. 31, no. 6, pp. 35–44, Nov 2014.

[22] M. C. Luizelli, L. R. Bays, L. S. Buriol, M. P. Barcellos, and L. P. Gaspary, "Piecing Together the NFV Provisioning Puzzle: Efficient Placement and Chaining of Virtual Network Functions," in *IFIP/IEEE International Symposium on Integrated Network Management (IM)*, May 2015, pp. 98–106.

[23] IBM Software, "Efficient modeling with the IBM ILOG CPLEX Optimization Studio," ILOG Optimization and Analytical Decision Support Solutions, Somers, NY, Tech. Rep. WSW14059-USEN-02, August 2010, White Paper. [Online]. Available: https://www.ibm.com/software/commerce/optimization/cplex-optimizer/