IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal
SPECIAL SECTION ON ROADMAP TO 5G: RISING TO THE CHALLENGE

Received February 18, 2019, accepted March 27, 2019, date of publication April 2, 2019, date of current version April 18, 2019.
Digital Object Identifier 10.1109/ACCESS.2019.2908990

Optimizing the Cloud Resources, Bandwidth and Deployment Costs in Multi-Providers Network Function Virtualization Environment

VINCENZO ERAMO [1] **AND FRANCESCO GIACINTO LAVACCA** [2]
[1]DIET, Sapienza University of Rome, 00184 Rome, Italy
[2]Fondazione Ugo Bordoni, 00161 Rome, Italy
Corresponding author: Vincenzo Eramo (vincenzo.eramo@uniroma1.it)

ABSTRACT The introduction of network function virtualization (NFV) leads to a new business model in which the Telecommunication Service Provider needs to rent cloud resources to infrastructure provider (InP) at prices as low as possible. Lowest prices can be achieved if the cloud resources can be rented in advance by allocating long-term virtual machines (VM). This is in contrast with the short-term VMs that are rented on demand and have higher costs. For this reason, we propose a proactive solution in which the cloud resource rent is planned in advance based on peak traffic knowledge. We illustrate the problem of determining the cloud resources in cloud infrastructures managed by different InPs and so as to minimize the sum of cloud resource, bandwidth and deployment costs. We formulate an integer linear problem (ILP) and due to its complexity, we introduce an efficient heuristic approach allowing for a remarkable computational complexity reduction. We compare our solution to a reactive solution in which the cloud resources are rented on demand and dimensioned according to the current traffic. Though the proposed proactive solution needs more cloud and bandwidth resources due to its peak allocation, its total resources cost may be lower than the one achieved when a reactive solution is applied. That is a consequence of the higher cost of short-term VMs. For instance, when a reactive solution is applied with traffic variation times of ten minutes, our proactive solution allows for lower total costs when the long-term VM rent is lower than the short-term VM one by 33%.

INDEX TERMS Network function virtualization, cloud infrastructure, short-term virtual machine, Viterbi algorithm.

I. INTRODUCTION

The 5G networks are envisioned to provide a flexible, scalable, agile and programmable network platform over which different services with varying requirements can be deployed and managed within strict performance bounds. Innovative concepts and techniques are being developed to power the 5G mobile networks. Two key technology enablers for realizing 5G networks are Network Function Virtualization and Software Defined Networking technologies [1].

Network Function Virtualization (NFV) is a new paradigm and employs cloud infrastructures to support telecommunication services. The service functions, also referred to as Virtual Network Functions (VNF) are executed in Virtual Machines hosted in geographically-distributed data centers [2]. In particular, NFV leverages the virtualization technology and allows network functions to be placed anywhere following an on-demand installation process. Thus, service providers can now offer highly specialized network services tailored to the end users' needs, without having to increase their capital investments for acquiring and installing specialized hardware devices and middleboxes. The benefits of NFV are multifold and range from the reductions in capital and operating expenditure to reducing the time to market of new services [3].

To support the NFV technology both ETSI [4], [5] and IETF [6] have been defining novel network architectures able to allocate resources for Virtualized Network Function (VNF) as well as to manage and orchestrate NFV to support services. In particular the service is represented by a Service Function Chain (SFC) [6] that is a set of VNFs that have to be executed according to a given order. Any VNF is run on a VNF Instance (VNFI) implemented with one Virtual Machine (VM) to which resources (cores, RAM memory,

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

disk memory) are allocated to execute a VNF of a given type (e.g., a virtual firewall, or a load balancer) [7], [8]

The research activity on NFV has been focusing on both the problem of SFC routing and choosing the servers in which the VNFIs are executed. Some performance indexes are considered as the number of SFCs accepted [9], [10], the server power consumption [11], [14], the bandwidth used to interconnect the VNFIs [15], [17], the cloud resource amount needed to activate the VNFIs [18], [19] and so on. Most of the papers analyze the problem in the static traffic scenario in which the SFC requests and their offered bandwidth are given [11]. Some solutions have been proposing in dynamic traffic scenario in which the SFCs have a given duration and/or their bandwidth and composition vary over the time. In these cases the proposed solutions have to take into account the network reconfiguration costs [20], [21] needed to change the routing of the SFC, the number of VNFIs activated and the servers in which the VNFIs are hosted. The reconfiguration costs can be characterized by the QoS degradation [11] and/or the energy consumption [12], [22] occurring during the migration of a VNFI, the deployment cost charged by the cloud provider to upload the VNF software into the Virtual Machine [21] and so on.

Most of proposed reconfiguration techniques are reactive that is they are activated when significant traffic changes are measured and this may lead to a delay in the application of the technique and a QoS degradation consequent. A proactive approach is proposed in [23] in which the authors aim to effectively estimate upcoming traffic rates and adjust the cloud resource allocation a priori, for flow service quality assurance and resource cost minimization; they adapt online learning techniques for predicting future SFC workloads. All of these solutions are based on low allocated cloud and bandwidth resource reconfiguration times with the consequence of a QoS degradation due to re-routing of the traffic flows; this degradation has been already investigated in [11] where a reconfiguration policy has been proposed to limit the bit loss occurring when the resources are re-allocated.

The introduction of NFV leads to a new business model with two main players [24]: the Infrastructure Provider (InP) owner of the cloud resources and the Telecommunication Service Provider (TSP) that provides user services, rents cloud resources from one or more than one InP and interconnects them with appropriate bandwidth links. The two players may be distinct and the TSP needs to rent cloud resources at the possible lowest price. That is possible if the cloud resources are rented in advance as long-term Virtual Machine (VM) instances [23], [25], [26] with respect to the case in which they are requested on demand as short-term VM instances [23].

In this paper we propose a proactive reconfiguration policy that starts from the assumptions that the nature of the traffic is cycle-stationary [27] and a TSP may be able to estimate the peak traffic in each stationary interval. The bandwidth and cloud resources are allocated according to this peak traffic in each stationary interval [27]. This allocation leads to an over-dimensioning and to a consequent waste of resources during the stationary interval but provides the TSP to know in advance the resource amount to be asked to the InP and consequently the possibility of renting long-term VM instances that are less expensive. Furthermore the proposed reconfiguration policy is based on times between reconfigurations higher than the RS solutions with the consequent advantages in lower QoS degradation.

The main contributions of the paper are: i) the proposal of a strategy for the allocation of long-term VM instances based on the knowledge of the peak traffic during the stationary intervals and with the objective of minimizing the sum of the costs of bandwidth, cloud and VNF deployment; ii) the comparison of the proposed strategy with the one in which the allocated resources are adapted to the current traffic but the cloud resources are requested on demand by renting short-term VM instances. For the first contribution we formulate the optimal allocation problem and propose an heuristic based on the Viterbi algorithm that lowers the computation times. We evaluate the effectiveness of the proposed strategy as a function of the traffic profile and the bandwidth, cloud and deployment cost unit. The rest of this paper is organized as follows. The main related works are discussed in Section II. The deployment cost aware cloud and bandwidth resource reconfiguration policy is illustrated in Section III. Cloud Infrastructure, network and traffic model are illustrated in Section IV. An Integer Linear Formulation (ILP) of the allocation problem of long-term VMs is shown in Section V. Section VI is devoted to propose an efficient heuristic to solve the problem with low computational complexity. Some numerical results are shown in Section VII to prove the effectiveness of the proposed solution and to compare it to the reactive one in which the resources are allocated according to the current traffic. Finally the main conclusions and future research items are mentioned in Section VIII.

II. RELATED WORK

The NFV resource allocation and routing problem has been studied in [28] by formulating it as a multi-commodity-chain network design problem on a cloud-augmented graph. The paper does not investigate dynamic traffic scenario and adopt a simple model in which the sharing of VNFIs among service functions and their maximum resource availability are not considered.

Some solutions have been proposing for bandwidth and cloud resource allocation in dynamic traffic scenario [29], [30]. Ghaznavi *et al.* [18], [31] design online algorithms to optimize the placement of VNF instances in response to on demand workload, considering the trade-off between bandwidth and server resource consumption. Eramo *et al.* [11], [12] introduce VM migration policies to save the server power consumption and by limiting the QoS degradation or the energy consumption occurring during the migrations. Wang *et al.* [20] propose a procedure to evaluate the number of VNFIs over the time with the objective to minimize the cloud resources used; their study does not

TABLE 1. Comparison between our proposal and the main related works.

Work	Dynamic Traffic Scenario	Cloud Resource Allocation in Multi-Provider Environment	Long Term Cloud Resource	Deployment Cost	Bandwidth Cost
Feng et al. [27]	No	No	No	No	No
Ghaznavi et al. [21,30]	Yes	No	No	No	Yes
Eramo et al. [14]	Yes	No	No	No	No
Wang et al. [34]; Liu et al [24]	Yes	No	No	Yes	No
Zhu et al. [34]; Eramo et al. [35]; Zhong et al. [36]	No	Yes	No	No	Yes
Zhang et al.[9]	Yes	No	Yes	No	No
Our Proposal	Yes	Yes	Yes	Yes	Yes

consider the cost of bandwidth resources. Liu *et al.* [21] study the problem of reallocation of VNFIs in a dynamic traffic scenario by constraining the number of reallocations so as to limit the deployment costs; they determine the number and the location of the VNFIs that are hosted in Virtual Machines (VM) equipped with a pre-determined number of cores dependent on the type of VNF supported by the VNFI.

The introduction of the NFV technology has led to a new business model with two main players [24], [32], [33]. The first one is the Infrastructure Provider (InP) whose the function is to deploy and manage the physical resources on which the virtual resources (i.e. Virtual Machine) may be provisioned. The physical resources are located in data centers and may be provisioned and rented through programming interfaces. Examples of InP could be public data centers such as those by Amazon or private and managed for instance by a telecommunication operator. The second player is the Telecommunications Service Provider (TSP): it rents the physical resources from one or more than one InP to execute the VNFs. It also provides to interconnect VNFs through bandwidth resources to create services to the end users. The TSP provides to rent the InPs, cloud resources at lowest price. The physical resources are located in distributed data centers and are interconnected by a transport infrastructure whose the TSP may be or not be the proprietary entity.

Few studies have been proposed in order to investigate solutions for the resource allocation and the cost optimization in Multi-Provider NFV environments. Sun *et al.* [34] propose and investigate an NFV scalable orchestration solution with the objective of minimizing the energy consumption. Wang *et al.* [35] study cloud and bandwidth resource allocation problems in geographically-distributed data centers interconnected by elastic optical networks; such a problem is also investigated in [36], [37] where the advantages of a provider cost aware resource allocation strategy are investigated. The scenario considered in [35], [36] suits very well to the business model of the NFV ecosystem but the authors consider a static traffic scenario only.

Most of the solutions proposed are reactive in nature by allocating resources according to the traffic demand. Conversely proactive solutions may be appropriate in NFV ecosystem in which the players TSP and InP are distinct and the TSP has to be able to predict its traffic demand so as to book in advance the cloud resources to the InP and consequently to save their cost. Zhang *et al.* [23] design an online, proactive VNF provisioning algorithm in which: i) a prediction of the traffic is performed; ii) the rent of both short-term and long-term VM instances is decided so as to support the offered traffic and where the short-term VMs are rent to absorb the traffic variability. The authors consider only cloud resources and they do not study the more general case of geographically-distributed data centers interconnected by a transport network.

In this paper we propose and investigate a solution for the cloud and bandwidth resource allocation and reconfiguration in NFV Multi-Provider environments [38]. We assume a knowledge of the peak traffic in Stationary Intervals of a classical cycle-stationary traffic scenario [39] in which the peak traffic is known following a estimation process out of the scope of this work; recently some promising traffic estimation techniques based on learning machine have been proposed and investigated [40]. The knowledge of the peak traffic allows for the use of long-term VM instances less expensive than the short-term ones. To save bandwidth and cloud resource, we propose a network reconfiguration policy applied at the beginning of each stationary interval in which the VNFIs are moved between data centers and their processing capacity is changed according to the expected traffic. The migrated VNFIs are decided so as to minimize the sum of the bandwidth, cloud and deployment costs.

Finally we compare in Table 1 our proposal and the main related works.

III. DEPLOYMENT COST AWARE CLOUD AND BANDWIDTH RESOURCE RECONFIGURATION POLICY

The allocation and reconfiguration policy is based on the assumption of a cycle-stationary traffic scenario [41], [43] organized in Stationary Intervals in which the peak traffic is known by means of an estimation process [40]. It determines in each SI: i) in which CIs to allocate the long term resources for the VNFIs; ii) how much resource to allocate to the VNFIs; iii) the network paths interconnecting the VNFIs and the bandwidth to be allocated for these paths. Its objective is to allocate and reconfigure resources so as to minimize the total cost given by the following cost components:

- Cloud resource rent cost: it is the rent cost of a VM instance supporting the VNFI with different composition of resources such as CPU cores, RAM and disk; each VM instance can be rented as short-term instance or long-term instance; in the case of short-term instances the VMs are requested on demand; in the case of long-term instances, the VMs are booked in advance and their provisioning cost is smaller than the one of short-term instances especially when the reservation occurs for a long time (1-3 years) [23].
- VNF deployment cost: it is the cost charged by an InP to copying the software of the VNF into the VM and launching it; when the software is launched a VNF instance (VNFI) is activated;
- bandwidth cost: it is the cost needed to interconnect the VNFIs; in particular we neglect the bandwidth cost for the interconnection of VNFIs located in a given data center while we consider the bandwidth cost to interconnect VNFIs located in geographically-distributed data centers.

A solution example is reported in Fig. 1 where the *CI*#1, *CI*#2, *CI*#3 and *CI*#4 Cloud Infrastructures, managed by four different InPs, are considered. Their cloud resources, consisting of 12 processing cores per each CI, can be interconnected by a network infrastructure. We assume that the processing core cost is charged by hour and it is given by $c^{core}_{CI_1} = 1\$/h$, $c^{core}_{CI_2} = 2\$/h$, $c^{core}_{CI_3} = 3\$/h$ and $c^{core}_{CI_4} = 4\$/h$ for *CI*#1, *CI*#2, *CI*#3 and *CI*#4 respectively. To simply our example we also assume that the bandwidth cost to interconnect two any VNFIs is given by $c^{ban} = 5\$/h$ when the two VNFIs are in two different CIs and 0 otherwise. Finally the deployment cost c^d is assumed equal to 1$ that is to say all of the times in which a VM executing a VNFI is restarted the InP charges the TSP by 1$. The traffic is assumed to be cycle-stationary with the profile reported in Fig. 1.b and composed by three SIs each one of time duration 1 hour and referred to as High Traffic (HT), Medium Traffic (MT) and Low Traffic (LT) respectively.

The VNFI graph [35] is composed by five VNFIs interconnected as indicated by the dashed arrows of Fig. 1.c.

The reconfiguration policy determines where the VNFIs have to be instantiated in the HT, MT and LT SIs respectively and how many cores the VNFIs need. The core allocations to the VNFIs and their locations are shown in Figs 1.c, 1.d, 1.e for the high, medium and low traffic scenario respectively. Notice how the objective of the policy, due to the low deployment costs (only 1$ needs to be paid when the execution of a VNFI is moved towards a new CI), is to minimize the sum of cloud and bandwidth resources costs in each SI. This minimization is achieved by both consolidating as much as possible the cloud resources towards the CIs with lowest cost cloud resources (CI#1 and CI#2 in the reported example) and respecting the constraint on the available core number in each CI. Finally the output of the reconfiguration policy is reported in Fig. 1.f. It specifies from which InPs and how many cloud resources (long-term VMs) the TSP will have to rent in each traffic interval. Finally we observe that the following costs are involved in the reported example:

- the sum of the cloud of bandwidth resource costs are equal to 62$, 38$ and 7$ for the HT, MT and LT SIs respectively. Each one of these costs is achieved by summing two terms; the first one characterizes the cloud resource cost and it is given by multiplying for each CI the total number of allocated cores and the price of core per hour; the second term is given by multiplying the number of network paths and the interconnection cost of any tuple of CIs.
- the total deployment cost in a cycle-stationary interval is given by 7$ because the cost to deploy one VNFI is assumed to be 1$ and 7 deployments are performed one, three and one in the configuration changes from HT SI to MT SI, from MT SI to LT SI and from LT SI to HT SI respectively);
- the total cost in a cycle-stationary interval is the one optimized by the proposed policy and in our example is given by 114$.

The cost components values are summarized in Fig. 1.g.

IV. CLOUD INFRASTRUCTURE, NETWORK AND TRAFFIC MODEL

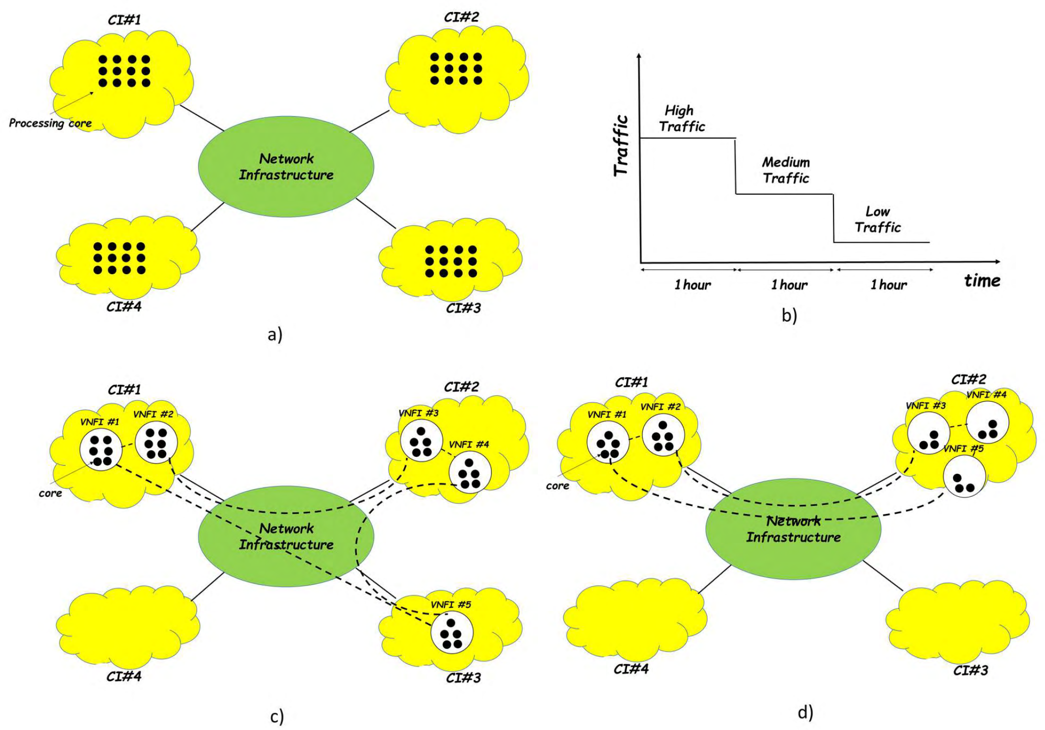
We illustrate the cloud infrastructure and network model in Subsection IV-A. The traffic model is illustrated in Subsection IV-B.

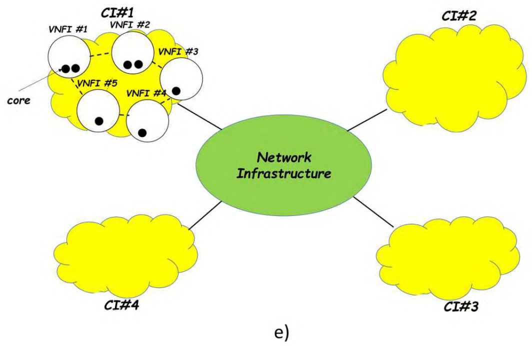
A. CLOUD INFRASTRUCTURE AND NETWORK MODEL

We represent the cloud infrastructures managed by the InPs and the network with the graph $\bar{G} = (\bar{V}, \bar{L})$ where $\bar{V} = \bar{V}_A \bigcup \bar{V}_S \bigcup \bar{V}_{CI}$ is a set of nodes that represents the access nodes in which the traffic requests, that is the Service Function Chains, are generated/terminated ($\bar{V}_A$), the network switches ($\bar{V}_S$) and the Cloud Infrastructures ($\bar{V}_{CI}$) managed by an InPs. $\bar{L}$ is the set of edges interconnecting the nodes of the set $\bar{V}$ and models the network links interconnecting the cloud infrastructures, access nodes and switches. The main Cloud Infrastructure and network Parameters are represented in Table 2.

We denote with $C_{\bar{e}}$ the capacity of the link $\bar{e} \in \bar{L}$, $L_{\bar{e}}$ is its length while c_b is the cost unit (Dollars/GbKm) of carrying 1 Gbit traffic on a link of 1 Km length.

The cloud infrastructure hosts servers providing the cloud resources needed to implement the network service. In this paper we consider only processing resources, though our study can be easily extended to the case in which memory and disk resources are also considered. We assume that the cloud infrastructure $\bar{v} \in \bar{V}_{CI}$ is equipped with a total number $N_{\bar{v}}$ of processing cores. Virtual Machines (VM) are activated in the cloud infrastructure to instantiate VNFIs supporting service functions as Network Address Translation (NAT), firewall (FW), Load Balancer (LB),. . . If we denote with F the number of service functions (SF), we assume that the





	VNFI #1	VNFI #2	VNFI #3	VNFI #4	VNFI #5
High Traffic SI	CI#1 6 cores	CI#1 6 cores	CI#2 5 cores	CI#2 5 cores	CI#3 5 cores
Medium Traffic SI	CI#1 5 cores	CI#1 5 cores	CI#2 3 cores	CI#2 3 cores	CI#2 3 cores
Low Traffic SI	CI#1 2 cores	CI#1 2 cores	CI#1 1 cores	CI#1 1 cores	CI#1 1 cores

f)

Cost				
Cloud and Bandwidth Resource			Deployment	Total
High Traffic SI	Medium Traffic SI	Low Traffic SI		
62$	38$	7$	7$	114$

g)

FIGURE 1. **An example of the reconfiguration policy. Four Cloud Infrastructures are interconnected by a transport network (a). A cycle-stationary traffic scenario is assumed with High Traffic (HT), Medium Traffic (MT) and Low Traffic (LT) Stationary Intervals (b). The location of the VNFIs supported by long-term VMs and their dimensioning in terms of number of cores is reported for HT (c), MT (d) and LT (e) Stationary Intervals. The output of the reconfiguration policy is reported (f) together with the cost components values (g).**

VNFI hosting the i-*th* $(i = 1, \cdots, F)$ SF is implemented with a software module providing the processing capacity $C_i^{pr,max}$ and requiring the allocation of n_i^c cores [14].

The traffic variations are handled with a vertical scaling, instead of horizontal [44], [45], in which the processing capacity of any VM is reduced with respect to the maximum one by lowering the number of cores allocated to the VM. The choice of a vertical scaling instead of a horizontal one is motivated by the need to reduce as much as possible the provisioning time of cloud resources. The time can be reduced by increasing/decreasing CPU, RAM and disk resources to the VMs rather than to instantiate/remove VMs [47]. The scaling operation can cause a degradation in Quality of Service in the downtime when the resources are re-allocated; the analysis of this degradation is out of the scope of this work but its cost impact may be studied by applying a procedure

TABLE 2. Cloud infrastructure and network parameters.

Parameter	Definition
$\bar{V_A}$	Set of access nodes
$\bar{V_S}$	Set of switch nodes
$\bar{L}$	Set of Links
$\bar{\Psi}$	Set of network paths
$L_{\bar{e}}$	Length of the link $\bar{e} \in \bar{L}$
$C_{\bar{e}}$	Capacity of the link $\bar{e} \in \bar{L}$
c_b	Bandwidth cost unit ($/GbKm)
$\bar{V_{CI}}$	Set of Cloud Infrastructures
$N_{\bar{v}}$	Number of cores in the CI $\bar{v} \in \bar{V_{CI}}$
F	Number of Service Functions (SF) types that can be executed in the CIs
$C_i^{pr,max}$	Maximum processing capacity of a i-th type VNFI
n_i^c	Number of cores to be allocated to a i-th type VNFI
$C_{i,j}^{pr}$	Processing capacity of the i-th type VNFI when j cores are allocated
$c_{\bar{v}}^{core}$	Cost unit of renting for one hour one processing core in the CI $\bar{v} \in \bar{V_{CI}}$
c_d^i	Deployment Cost of a VNFI supporting the i-th type SF

already investigated [11]. It consists in limiting the number of scaling operations so as to find a right compromise between the advantages in used resource costs and QoS degradation. We assume that for the i-th type VNFI, the processing capacities $C_{i,j}^{pr} = j\frac{C_i^{pr,max}}{n_i^c}$ $(j = 1, \cdots, n_i^c)$ can be provided requiring the allocation of j cores respectively.

Next we denote with $c_{\bar{v}}^{core}$ the cost unit ($/hour) of renting for one hour one processing core in the Cloud Infrastructure (CI) $\bar{v} \in \bar{V_{CI}}$. Finally we assume that any InP charges a deployment cost c_d^i $(i = 1, \cdots, F)$ to deploy the software implementing the i-th service function.

B. TRAFFIC MODEL

The traffic parameters are illustrated in Table 3. We assume cycle-stationary traffic with T Stationary Intervals (SI), each one of duration time T_s and corresponding a duration $T_{cs} = T * T_s$ of the cycle-stationary period. The TSP has to serve N traffic requests each one characterized by a 5-tuple $< \bar{v}_i, \bar{w}_i, M_i, \vec{s}_i, \vec{b}_i >$ $(i = 1, \cdots, N)$ where:

- $\bar{v}_i \in \bar{V_A}$ and $\bar{w}_i \in \bar{V_A}$ denote the originating and terminating access nodes the traffic request;
- M_i is the number of service functions to be executed according to a given order for the traffic request; we assume that the SFs belongs to a set of F types;
- $\vec{s}_i$ is a 1's or 0's matrix of size $M_i \times F$; the component $s_i(j, k)$ assumes the value 1 if the j-th SF to be executed is a k-th type SF;
- $\vec{b}_i$ is a vector of size T; the component $b_i(j)$ $(j = 0, \cdots, T-1)$ is the peak traffic of the i-th traffic request in the j-th SI; the TSP will ask for the InP the resource allocation on the basis of these peak values.

TABLE 3. Traffic parameters.

Parameter	Definition
N	Number of SFCs
T_s	Duration time of a Stationary Interval (SI)
T	Number of SIs in a cycle-stationary period
M_i	Number of Service Functions composing the offered i-th SFC
$s_i(j,k)$	It assumes the value 1 if the j-th SF to be executed is a k-th type SF; otherwise its value is zero
$b_i(j)$	It is the bandwidth requested by the i-th SFC in the j-th SI; otherwise its value is zero

Though the traffic is a stochastic process, we assume that the bandwidth value $b_i(j)$ requested by the i-th SFC $(i = 1, \cdots, N)$ in the j-th $(j = 0, \cdots, T-1)$ SI is deterministic; we assume that these values are α-percentiles of an estimated traffic distribution with values of α chosen so as to make sufficiently low the probability that the bandwidth requested is higher of the α-percentile. An example of determination of the bandwidth values in a real traffic scenario is provided in [46].

The introduced traffic model is general and allows for handling any traffic variation once that the parameters T_s and T are determined. For instance the choice of $T_s = 3$ hour, $T = 8$ and consequently $T_{cs} = 24$ hour will allow for the modeling of a typical daily traffic profile. When the duration T_{cs} of the cycle-stationary is fixed, the choice of the parameter T_s is established according to a right trade-off between resources over provisioning and deployment costs. In fact notice as the increase in T_s leads to higher predicted peak bandwidths $b_i(j)$ $(i = 1, \cdots, N; j = 0, \cdots, T-1)$ and consequently a higher over provisioning of cloud and bandwidth resources; on the other hand this increase leads to lower values of T and a reduction of the deployment costs.

V. INTEGER LINEAR PROGRAMMING FORMULATION

The objective is to determine in which CI to rent the cloud resources to execute the VNFIs and which network paths to use to interconnect them. The cloud resource allocation must be determined in each SI and it is accomplished so as to minimize the cloud resources and bandwidth rent costs and the deployment costs.

Before illustrating the optimal problem formulation we need to determine the VNFI graph [11], [35], [36]. Each VNFI executes an SF of a given type and can be shared among the SFCs offered. The VNFI graph is determined so as to include a sufficient number of VNFIs to support the maximum traffic offered in a cycle-stationary period. It is determined with the procedure illustrated in [36] and revised in Appendix A. The procedure is based on an algorithm referred to as SFC Routing and Cloud and Bandwidth resource Allocation (SRCBA) algorithm. To guarantee that a sufficient number of VNFIs is instantiated, the SRCBA algorithm is applied when the maximum traffic $\beta_i^{max} = \max_{1\leq j\leq T} b_i(j)$ $(i = 1, \cdots, N)$ offered for each SFC is considered. The inputs of the algorithm are: the cloud and bandwidth resources in the CI and network infrastructure, the SFCs offered and the bandwidth values β_i^{max} $(i = 1, \cdots, N)$, the network access nodes in which the SFCs are originated and terminated. The outputs of the algorithm are: the VNFI graph, an initial embedding of the VNFI graph into the CI and network infrastructure specifying in which CI each VNFI is placed and in which network path each virtual link is routed. The objective of the algorithm is the minimization of the sum of allocated cloud and bandwidth resource cost. The embedding is denoted as Γ^{MT}.

The VNFI graph $G = (V, L)$ is characterized by a set V of nodes and links L. The definition of the main VNFI graph parameters is reported in Table 4.

TABLE 4. VNFI graph parameters.

Parameter	Definition
V_A	Set of virtual nodes in which the SFCs are generated
V_{VI}	Set of VNFIs
$f_v^{(h)}$	Processing capacity requested by the VNFI $v \in V_{VI}$ in the h-*th* SI
$f_e^{(h)}$	Bandwidth requested by the virtual link $e \in L$ in the h-*th* SI
σ_{iv}	Binary parameter that assumes the 1's value if $v \in V_{VI}$ executes i-*th* type SFs
$\mu_{v\bar{v}}$	Binary parameter that assumes 1's value if $v \in V_{VI}$ is located in $\bar{v} \in \bar{V_A}$

The set V of nodes is the union of sets V_A and V_{VI} where:

- V_A is the set of virtual nodes in which the SFC requests are generated/terminated; this nodes are always hosted in a node of the set $\bar{V_A}$ of the access nodes; their locations are not an output of the optimization problem;
- V_{VI} is the set of VNFI nodes; each node is characterized by the type of SF it is able to execute; we assume that the processing resources of one VNFI node can be shared by SFs of the same type and belonging to different SFCs; the mode in which the SFs are assigned to any VNFI node is established by the SRCBA algorithm described in Appendix A; conversely the optimization problem determines in which nodes of the set $\bar{V_{CI}}$ these nodes must be hosted; the decision has to be taken in each SI.

The set L contains edges interconnecting the nodes of the set V and the optimization problem must determine in which network paths these edges must be supported. In particular if the nodes interconnected by the edges are located in different CIs, their interconnection involves a bandwidth consumption. Next the main parameters and variables of the optimization problem are described. We define the following parameters:

- σ_{iv} $(i = 1, \cdots, F; v \in V_{VI})$: it assumes the 1's value if the VNFI $v \in V_{VI}$ executes the i-*th* type SF; otherwise its value equals 0;
- $\mu_{v\bar{v}}$ $(v \in V_A; \bar{v} \in \bar{V_A})$: it assumes the 1's value if the access node $v \in V_A$ of the VNFI graph is starting/terminating from/to the access node $\bar{v} \in \bar{V_A}$ of the network infrastructure; otherwise its value equals 0;
- $f_v^{(h)}$ $(v \in V_{VI}; h = 0, 1, \cdots, T-1)$: it denotes the processing capacity (b/s) requested by the VNFI $v \in V_{VI}$ in the h-*th* SI;
- $f_e^{(h)}$ $(e \in L; h = 0, 1, \cdots, T-1)$: it denotes the bandwidth (b/s) requested by the link virtual $e \in L$ in the h-*th* SI;

The meaning of the optimization variables and outputs are also reported in Table 5.

TABLE 5. Optimization variables and outputs.

Variable and Output	Definition
$x_{v\bar{v}}^{(h)}$	Binary variable that assumes the 1's value if $v \in V_{VI}$ executes in $\bar{v} \in \bar{V_{VI}}$ in the h-*th* SI
$y_{e\bar{p}}^{(h)}$	Binary variable that assumes the 1's value if virtual link e is routed on the path $\bar{p}$ in the h-*th* SI
$f_{\bar{v}}^{(h)}$	Binary variable that assumes the 1's value if $\bar{v} \in \bar{V_{VI}}$ has changed the execution CI
C_{tot}	Total cost expressed by Eq. (8)
C_{CR}	Cloud resource cost expressed by Eq. (9)
C_{BW}	Bandwidth resource cost expressed by Eq. (10)
C_D	Deployment cost expressed by Eq. (11)

We also assume that for each tuple of nodes of the CI and network infrastructure we have pre-calculated the set of the P shortest paths; this paths are stored in the set $\bar{\Upsilon}$. The binary parameter $\delta_{\bar{e}\bar{p}}$ $(\bar{e} \in \bar{L}; \bar{p} \in \bar{\Upsilon})$ is also introduced and it assumes 1's values when the link $\bar{e} \in \bar{L}$ belongs to the network path $\bar{p} \in \bar{\Upsilon}$. The functions $S(e) \in V$ and $D(e) \in V$ denote the origin and destination nodes of the virtual link $e \in L$, conversely the functions $\bar{S}(\bar{p}) \in \bar{V}$ and $\bar{D}(\bar{p}) \in \bar{V}$ denotes the origin and destination nodes of the network path $\bar{p} \in \bar{\Upsilon}$

Three binary optimization variables are introduced:

- $x_{v\bar{v}}^{(h)}$ $(v \in V_{VI}; \bar{v} \in \bar{V_{CI}}; h = 0, 1, \cdots, T-1)$: it assumes 1's value if the VNFI $v \in V_{VI}$ of the VNFI graph executes in the CI $\bar{v} \in \bar{V_{CI}}$ in the h-*th* SI; otherwise its value equals 0;
- $y_{e\bar{p}}^{(h)}$ $(e \in L; \bar{p} \in \bar{\Upsilon}, h = 0, 1, \cdots, T-1)$: it assumes the 1's value if the virtual link $e \in L$ of the VNFI graph

is routed on the path $\bar{p} \in \bar{\Upsilon}$ in the h-*th* SI; otherwise its value equals 0;

- $z_v^{(h)}$ ($v \in V_{VI}; h = 0, 1, \cdots, T-1$): it assumes 1's value in the h-*th* SI if the VNFI $v \in V_{VI}$ of the VNFI graph has changed the CI in which it executes; otherwise its value equals 0; this variable is introduced to linearize the optimization problem.

In defining the optimization problem, we model the resource consumption according to classical assumptions [11], [14], [18], [31]; in particular the two main constraints on the cloud and bandwidth resources consumption establish that: i) the sum of the processing capacities of the VNFIs executing in any CI is lower than the available one in the CI; the sum of the bandwidths of the VNFI links routed in a network link is lower than the total link capacity.

Next we report all of the constraints of the optimization problem.

$$\sum_{\bar{v} \in \bar{V_{CI}}} x_{v\bar{v}}^{(h)} = 1 \quad v \in V_{VI}; \quad h \in [0..T-1] \tag{1}$$

The constraint (1) establishes that a VNFI $v \in V_{VI}$ has to be executed on only one CI $\bar{v} \in \bar{V_{CI}}$ in the h-*th* SI. Notice that constraint (1) does not prevent that VNFI of the same type (i.e. a Firewall) from being executed in more than one CI. In fact the set V_{VI} of the graph VNFI can contain more than one VNFI of a given type depending on the bandwidth requests of the corresponding Service Function. These different VNFIs may be executed on different CIs once the optimization problem is solved.

$$\sum_{\bar{p} \in \bar{\Upsilon}} y_{e\bar{p}}^{(h)} = 1 \quad e \in L; \quad h \in [0..T-1] \tag{2}$$

The constraint (2) establishes that a virtual link $e \in L$ has to be routed on only one network path $\bar{p} \in \bar{\Upsilon}$ in the h-*th* SI.

$$\sum_{v \in V_{VI}} x_{v\bar{v}}^{(h)} \left\lceil \frac{f_v^{(h)}}{\sum_{k=1}^{F} \sigma_{kv} \frac{C_k^{pr,max}}{n_k^c}} \right\rceil \leq N_{\bar{v}} \quad \bar{v} \in \bar{V_{CI}}; \quad h \in [0..T-1] \tag{3}$$

The constraint (3) guarantees that core resources are available for all of the VNFIs executing in a CI; it guarantees that the VNFIs placed in a CI $\bar{v} \in \bar{V_{CI}}$ in the h-*th* SI require a total amount of cloud resource lower than the one allocated to the CI. In particular notice how the ceiling operator provides the number of cores to be allocated to the VNFI $v \in V_{VI}$.

$$\sum_{e \in L} f_e^{(h)} \sum_{\bar{p} \in \bar{\Upsilon}} y_{e\bar{p}}^{(h)} \delta_{\bar{e}\bar{p}} \leq C_{\bar{e}} \quad \bar{e} \in \bar{L}; \quad h \in [0..T-1] \tag{4}$$

The constraint (4) guarantees that any link $\bar{e} \in \bar{L}$ is not overloaded in the h-*th* SI.

$$z_v^{(h)} \geq x_{v\bar{v}}^{(h)} - x_{v\bar{v}}^{(h-1) \bmod T} \quad v \in V_{VI}; \; \bar{v} \in \bar{V_{CI}}; \quad h \in [0..T-1] \tag{5}$$

The constraint (5) establishes that a VNFI $v \in V_{VI}$ has been moved in the h-*th* SI when it executes in the CI $\bar{v} \in \bar{V_{CI}}$ in the h-*th* SI but it does not in the $((h-1) \bmod T - th)$ SI.

Finally last constraints guarantees for $e \in L$, $\bar{p} \in \bar{\Upsilon}$ and $h \in [0..T-1]$ the following conditions:

$$y_{e\bar{p}}^{(h)} \leq \begin{cases} x_{S(e)\bar{S}(\bar{p})}^{(h)} & \text{if } S(e) \in V_{VI} \\ \mu_{S(e)\bar{S}(\bar{p})} & \text{if } S(e) \in V_A \end{cases} \tag{6}$$

$$y_{e\bar{p}}^{(h)} \leq \begin{cases} x_{D(e)\bar{D}(\bar{p})}^{(h)} & \text{if } D(e) \in V_{VI} \\ \mu_{D(e)\bar{D}(\bar{p})} & \text{if } D(e) \in V_A \end{cases} \tag{7}$$

The constraints (6)-(7) guarantee that the origin and destination nodes of any virtual link are located in the start and end nodes in the cloud and network infrastructure of the corresponding path in which the virtual link is routed.

The objective function C_{tot} to be minimized is composed by the following three terms:

$$C_{tot} = C_{CR} + C_{BW} + C_D \tag{8}$$

wherein:

- the term C_{CR} is the cost component of the cloud resource used in a cycle-stationary interval; if T_s is the duration of any SI, its expression is given by:

$$C_{CR} = T_s \sum_{h=0}^{T-1} \sum_{\bar{v} \in \bar{V_{CI}}} c_{\bar{v}}^{core} \sum_{v \in V_{VI}} x_{v\bar{v}}^{(h)} \left\lceil \frac{f_v^{(h)}}{\sum_{k=1}^{F} \sigma_{kv} \frac{C_k^{pr,max}}{n_k^c}} \right\rceil \tag{9}$$

- the term C_{BW} is the cost component of the bandwidth used to interconnect the VNFIs; its expression is given by:

$$C_{BW} = T_s c_b \sum_{h=0}^{T-1} \sum_{\bar{e} \in \bar{L}} \sum_{e \in L} f_e^{(h)} \sum_{\bar{p} \in \bar{\Upsilon}} y_{e\bar{p}}^{(h)} L_{\bar{e}} \delta_{\bar{e}\bar{p}} \tag{10}$$

- the term C_D is the deployment cost charged by an InP every time an VNFI is activated; its expression is given by:

$$C_D = \sum_{h=0}^{T-1} \sum_{i=1}^{F} \sum_{v \in V_{VI}} c_d^i z_v^{(h)} \tag{11}$$

To prove that the optimization problem is strongly NP-hard it is sufficient to consider the case of one stationary interval ($T = 1$) and infinite link bandwidth. In such a case the introduced problem reduces to the Multi-dimensional Bin Packing Problem that Garvey and Graham [48] have shown to be strongly NP-hard. Due to this high complexity, we propose two heuristics to be jointly applied to determine in each SI one embedding of the graph VNFI into the cloud and network infrastructures so as to minimize the total cost.

VI. ALGORITHMS FOR THE DEPLOYMENT COSTS AWARE CLOUD AND BANDWIDTH RESOURCE RECONFIGURATION

The high complexity of the optimization problem leads us to define some heuristics for the cloud resource and bandwidth reconfiguration with computational complexity as low as possible. We develop a two-steps solution in which the following actions are performed:

- the first step aims at evaluating a least cloud resource and bandwidth cost embedding of the VNFI graph in the cloud and network infrastructure in each SI and under the traffic scenario of that SI; the Least Cloud resource and Bandwidth Cost (LCBC) heuristic has been developed to evaluate these embeddings; its operation mode is described in Subsection VI-A
- the second step aims at defining an allocation and reconfiguration policy that takes into account the deployment costs; the policy aims at determining which embedding, chosen among the ones evaluated in the first phase, to apply in each SI; the objective of the policy will be the one of minimizing over the cycle-stationary interval the sum of the three cost components: cloud and bandwidth resource cost and deployment cost; in particular when the deployment costs are low, the policy will apply in each SI the least cloud resource and bandwidth cost embedding evaluated for the traffic condition of the SI; conversely for high deployment costs, the policy will change as little as possible the embeddings applied in the SIs so as to reduce the deployment costs; the Deployment Costs Aware (DCA) heuristic, described in Subsection VI-B defines the operation mode of the reconfiguration policy.

A. LEAST CLOUD RESOURCE AND BANDWIDTH COST (LCBC) HEURISTIC

The LCBC heuristic allows for the evaluation in each SI of an embedding of the VNFI graph into the CI and network infrastructure so as to minimize the cloud resource and bandwidth costs. The LCBC heuristic is applied in each SI and tries to consolidate the cloud and bandwidth resources starting from the maximum traffic embedding Γ^{MT} mentioned in Section V.

Next we describe the LCBC heuristic when it is applied in the h-*th* SI.

The main steps of the LCBC heuristic are the following: i) it starts from Γ^{MT} and according to the resource reduction requested by the nodes and links of the graph VNFI during the h-*th* SI, tries to move the execution of the VNFIs towards the lowest cost CI and by reducing as much as possible the CI interconnection links and consequently the used bandwidth; ii) if N_{CI} is the size of the set $\bar{V_{CI}}$, the LCBC heuristic evaluates N_{CI} new embeddings obtained from the embedding Γ^{MT} by moving the VNFIs between CIs; iii) the costs of the N_{CI} embeddings are evaluated and the least cloud resource and bandwidth cost one is chosen.

Each of the N_{CI} embeddings is evaluated as follows. Two edges $\bar{V_{CI}^a}$ and $\bar{V_{CI}^b}$ are defined and referred to as hosting and hosted sets respectively. The sets $\bar{V_{CI}^a}$ and $\bar{V_{CI}^b}$ form a partition of the set $\bar{V_{CI}}$. Next we will specify how both the partition is chosen and its choice characterizes the embedding. The LCBC algorithm tries moving VNFIs executing in CIs belonging to the set $\bar{V_{CI}^b}$ towards CIs belonging to $\bar{V_{CI}^a}$. The VNFIs to be moved are selected from the nodes $\bar{v} \in \bar{V_{CI}^b}$ ordered according to decreasing values of the index $d_{\bar{v}}^{\bar{V_{CI}^a}}$ characterizing the cost advantage in switching off the CI $\bar{v}$ when its VNFIs are moved towards the CIs belonging to the set $\bar{V_{CI}^a}$. The advantages are twofold: i) the one of saving the cloud resources used in the node $\bar{v}$; ii) the one of saving bandwidth for the interconnection of the VNFIs in $\bar{v}$ to the VNFIs located in CIs belonging to the set $\bar{V_{CI}^a}$. For this reason $d_{\bar{v}}^{\bar{V_{CI}^a}}$ can be expressed as follows:

$$d_{\bar{v}}^{\bar{V_{CI}^a}} = c_{\bar{v}}^{core} N_{\bar{v}}^{core,u,(h)} T_s + \sum_{\bar{w} \in \bar{V_{CI}^a}} c_b L_{\bar{v}\bar{w}} T_{\bar{v}\bar{w}}^{(h)} T_s \quad (12)$$

wherein:

- $N_{\bar{v}}^{core,u,(h)}$ is the number of cores used by CI $\bar{v}$ in h-*th* SI;
- $L_{\bar{v}\bar{w}}$ is the length in Km of the paths between $\bar{v}$ and $\bar{w}$;
- $T_{\bar{v}\bar{w}}^{(h)}$ is the total traffic (Gbps) carried by the link interconnecting $\bar{v}$ and $\bar{w}$ in h-*th* SI.

To reduce as much as possible the computational complexity, the N_{CI} partitions of the LCBC heuristic are generated in an incremental way. Initially the set $\bar{V_{CI}^a}$ is composed by the least cloud resource cost CI while the remaining CIs are inserted in the set $\bar{V_{CI}^b}$. At each step the partition is changed by moving one CI from $\bar{V_{CI}^b}$ to $\bar{V_{CI}^a}$. The algorithm selects the node $\bar{v} \in \bar{V_{CI}^b}$ with the smaller value of the index $e_{\bar{v}}^{\bar{V_{CI}^a}}$ charactering the disadvantage of maintaining turned on the node CI $\bar{v}$ when the CIs in which to consolidate the VNFIs are in the set $\bar{V_{CI}^a}$. This disadvantage is characterized by: i) the high cost of the cloud resources per processed traffic Gbit of the node CI $\bar{v}$; ii) the high cost of carrying one traffic Gbit from the node CI $\bar{v}$ to the other nodes CIs belonging to the set $\bar{V_{CI}^a}$. For this reason we can express $e_{\bar{v}}^{\bar{V_{CI}^a}}$ as follows:

$$e_{\bar{v}}^{\bar{V_{CI}^a}} = \frac{1}{F} \sum_{i=1}^{F} \frac{c_{\bar{v}}^{core} n_i^c}{C_i^{pr,max}} + \sum_{\bar{w} \in \bar{V_{CI}^a}} c_b L_{\bar{v}\bar{w}} \quad (13)$$

where the first term of the second hand of expression (13) characterizes the processing cost per one traffic Gbit in the CI $\bar{v}$; conversely the second term characterizes the average cost of carrying one traffic Gbit between the CIs $\bar{v}$ and the other CIs located in the set $\bar{V_{CI}^a}$.

The main steps of the LCBC heuristic are reported in Algorithm 1. The input is the maximum traffic embedding Γ^{MT} (line 1) and the outputs are the determined embedding $\Gamma^{(h)}$ and it's the cloud resource and bandwidth cost $C^{(h)}$ in the h-*th* SI (line 24).

Algorithm 1 LCBC Algorithm

1: **Input:** Maximum Traffic Embedding Γ^{MT}
2: ***/*Initialization Phase*/***
3: $\bar{V^a_{CI}} = \emptyset$; $\bar{V^b_{CI}} = \bar{V_{CI}}$;
4: $\Gamma^{(h)} = \Gamma^{MT}$; $C^{(h)} = \infty$
5: **while** $\bar{V^b_{CI}} \neq \emptyset$ **do**
6: $\quad\Gamma = \Gamma^{MT}$;
7: $\quad$**Choose** $\bar{v} = \arg\min_{\bar{w} \in \bar{V^b_{CI}}} e_{\bar{w}}^{\bar{V^a_{CI}}}$
8: $\quad\bar{V^a_{CI}} = \bar{V^a_{CI}} \cup \{\bar{v}\}$; $\bar{V^b_{CI}} = \bar{V^b_{CI}} - \{\bar{v}\}$
9: $\quad\chi = \bar{V^b_{CI}}$;
10: $\quad$**while** $\chi \neq \emptyset$ **do**
11: $\quad\quad$**Choose** $\bar{v} = \arg\max_{\bar{w} \in \bar{V^b_{CI}}} d_{\bar{w}}^{\bar{V^a_{CI}}}$
12: $\quad\quad\chi = \chi - \{\bar{v}\}$
13: $\quad\quad$**Determine** all of the VNFIs in $\bar{v}$ that can be mapped in any CI in $\bar{V^a_{CI}}$ according to the cloud resource availability
14: $\quad\quad$**if** Virtual Links of involved VNFIs can be routed **then**
15: $\quad\quad\quad$**Update** the embedding Γ
16: $\quad\quad$**end if**
17: $\quad$**end while**
18: $\quad$**Evaluate** the cloud resource and bandwidth cost C of the new embedding
19: $\quad$**if** $C \leq C^{(h)}$ **then**
20: $\quad\quad C^{(h)} = C$
21: $\quad\quad\Gamma^{(h)} = \Gamma$
22: $\quad$**end if**
23: **end while**
24: **Output:** $\Gamma^{(h)}$, $C^{(h)}$

In the Initialization Phase the sets $\bar{V^a_{CI}}$ and $\bar{V^b_{CI}}$ are initialized (line 3). At each step (line 5) the partition is changed by moving the CI $\bar{v}$, chosen according to the minimization of the index (13) (line 7), from $\bar{V^b_{CI}}$ to $\bar{V^a_{CI}}$ (line 8). Next LCBC heuristic tries moving the VNFIs from $\bar{V^b_{CI}}$ to $\bar{V^a_{CI}}$ and the CIs are selected in decreasing order of the index expressed by (12) (line 11). Then the VNFIs in the CIs $\bar{v} \in \bar{V^b_{CI}}$ that can be moved in any CI of $\bar{V^a_{CI}}$ are determined by taking into account the cloud resource availability (line 13). If bandwidth is available to move the VNFIs (line 14) then the embedding Γ is updated (line 15). Notice as the checks performed in lines (13-14) guarantee the determination of an embedding in which no more that the available cloud and bandwidth resource amount is used. When all of the nodes $\bar{v} \in \bar{V^b_{CI}}$ have been considered, the cost of the final embedding of the considered partition is evaluated (line 18) and if it is lower than the current one (line 19), both the cost and the embedding are updated (lines 20-21).

As far as the LCBC computational complexity is concerned, we can observe that: i) N_{CI} embeddings are evaluated; ii) the CIs of the set $\bar{V^b_{CI}}$ are selected in decreasing order of the parameter value expressed by (12); iii) if N_{core}^{max} is the maximum number of cores in any CI then it may be needed move at most N_{core}^{max} VNFIs; iv) the displacement of any VNFI needs a re-routing of virtual links and the application of the Dijkstra algorithm with complexity $O((N_{CI} + N_A + N_S) log N_L)$, being N_A the number of access nodes, N_S the number of switches and N_L the number of network links. According to these remarks we can conclude that the computational complexity of the LCBC heuristic is $O(N_{CI}^3 N_{core}^{max} (N_{CI} + N_A + N_S) log N_L)$.

B. DEPLOYMENT COSTS AWARE (DCA) HEURISTIC

The objective of the Deployment Costs Aware (DCA) heuristic is to maintain low the deployment costs. Instead of applying the least cloud resource and bandwidth cost embedding $\Gamma^{(h)}$ ($h = 0, \cdots, T-1$) in the h-*th* SI, the DCA heuristic tries to limit the embedding changes when a new SI occurs so as to reduce the deployment costs. The output of the DCA heuristic is the set $\Sigma^{opt} = \{\Psi_0, \Psi_1, \cdots, \Psi_{T-1}\}$ where Ψ_i is an embedding chosen among the ones evaluated by the LCBC heuristic that is in the set $\Theta = \{\Gamma^{(0)}, \Gamma^{(1)}, \cdots, \Gamma^{(T-1)}\}$. The set Σ^{opt} is chosen so as to minimize the sum of the cloud resource and bandwidth cost and the deployment cost. We show how the solution of the heuristic consists in evaluating a least cost cyclic path in a multi-stage graph. A low computational complexity methodology to solve the problem and based on the Viterbi algorithm is described in [14]- [12]. The determination of the set Σ^{opt} in the DCA heuristic can be accomplished by solving a least cost cyclic path problem in the multi-stage graph $\mathcal{G}^{\mathcal{DCA}}$ organized in T stages numbered from 0 to $T-1$. The i-th stage contains n_i nodes, whose the generic one is named $v_{i,j}$ ($i = 0, \cdots, T-1; j = 1, \cdots, n_i$). The number n_i of nodes in each stage is equal to the number of embeddings belonging to the set Θ and admissible for the traffic offered in the i-*th* SI. In particular an embedding is said to be admissible for a traffic state when the instantiation of the VNFIs in the CIs and the routing of the virtual links in the network do not lead to overcome the CI processing capacities and the link bandwidth respectively. A node $v_{i,j}$ of the multi-stage graph is characterized by:

- an embedding belonging to the set Θ and admissible for the traffic offered in the i-*th* SI;
- the sum of cloud resource and bandwidth costs when the embedding is applied.

An edge of the multi-stage graph characterizes an embedding change due to the displacement of VNFIs. The edge is labeled with the deployment cost that the embedding change involves due to the activation of the VNFIs in the hosting CIs.

We provide an example of application of the DCA heuristic considering the cycle-stationary traffic scenario of Fig. 1 and composed by the HT ($h = 0$, MT ($h = 1$) and LT ($h = 2$) SIs. We assume the CI and network infrastructure of Fig. 1 in which the CIs are equipped with 12 cores. We also assume the same core and bandwidth cost values of the example described in Section III. In this case the least cost embedding are the ones illustrated in Figs 1.c, 1.d and 1.e for $h = 0, 1, 2$ and referred to as $\Gamma^{(0)}$, $\Gamma^{(1)}$ and $\Gamma^{(2)}$ respectively. The costs

	h=0	h=1	h=2
$\Gamma^{(0)}$	62	46	26
$\Gamma^{(1)}$	—	38	20
$\Gamma^{(2)}$	—	—	7

a)

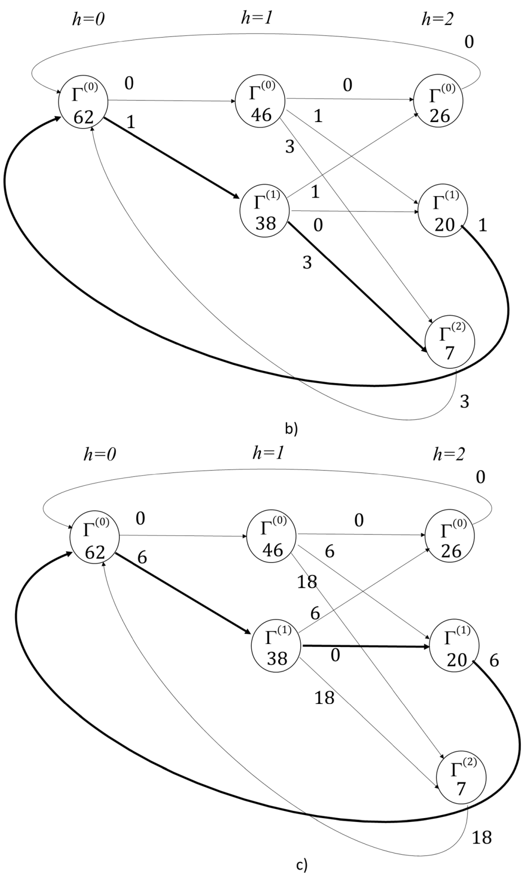


FIGURE 2. An example of application of the DCA heuristic. Embedding costs (a) and graph $\mathcal{G}^{\mathcal{DCA}}$ reported in the cases of deployment cost c_d equal to 1$ (b) and 6$ (c); the core and bandwidth cost values of the example reported in Section III are considered. The thick lines path reported in the graphs identifies the sequence of the embeddings to be applied to minimize the total cost.

of the embeddings in each SI are reported in Fig. 2.a. In that SI where the cost is not reported for an embedding, it means that the embedding is not admissible for the peak traffic offered in that SI. We evaluate the policy of the LCBC/DCA heuristic in the cases of deployment cost c_d equal to 1$ and 6$. Notice as the first choice of the deployment cost ($c_d = 1$$) corresponds to the chosen value in the example reported in Section III. We report the multi-stage graphs in Figs 2.b and 2.c for the case of deployment cost equal to 1$ and 6$ respectively. The number of admissible embeddings in the h-*th* SI equals 1, 2 and 3 for $h = 0, 1, 2$ respectively. Inside the nodes we report the corresponding mapping and the sum of the cloud resource and bandwidth costs when the embedding is applied. Each link is labeled with the deployment cost needed to perform the corresponding embedding change. Once built the multi-stage graph, the least cost cyclic path can be computed to achieve the embeddings to be applied in each SI so as to minimize the total cost in a cycle-stationary period. The thick lines paths reported in Figs 2.b and 2.c identify the embeddings to be applied to minimize the total cost for the cases of deployment costs equal to 1$ and 6$ respectively. From Fig. 2.b we can notice how in the case of low deployment cost ($c_d = 1$$) the chosen policy is the one characterized by the mappings $\Gamma^{(0)}$, $\Gamma^{(1)}$ and $\Gamma^{(2)}$ that is the least cost embeddings. Conversely from Fig. 2.c we can observe that in the case of high deployment costs ($c_d = 6$$) the chosen policy is the one characterized by the embeddings $\Gamma^{(0)}$, $\Gamma^{(1)}$ and $\Gamma^{(1)}$ that is to say that in the Low Traffic SI the least cost embedding is not chosen to limit the deployment cost.

Finally we conclude by discussing about the complexity of the DCA heuristic. Because the determination of a least cost cyclic path can be reduced to the one of a shortest path, we can affirm that the complexity is the one of the Dijkstra algorithm that in the worst case is applied in a graph with T^2 nodes and T^3 links. It follows that if the candidate list in the Dijkstra algorithm is implemented with a binary heap [49], the computational complexity of the DCA heuristic is $O(T^3 logT)$.

VII. NUMERICAL RESULTS

The section is devoted to evaluate the effectiveness of the LCBC/DCA heuristics by comparing their results with the ones of the ILP problem described in Section V and solved with the CPLEX solver. Next we verify when our approach based on the peak traffic knowledge is effective and allow for cost saving with respect to a reactive solution [20] in which the cloud and bandwidth resources are reconfigured as soon as a traffic variation occurs. The cost saving of our solution is due to the possibility of allocating in advance cloud resources and using long-term Virtual Machines, less expensive than short-term ones.

We consider four Service Functions: Firewall (FW), Intrusion Detection System (IDS), Network Address Translator (NAT) and Proxy. We consider four types of SFCs. The first one composed by a Firewall (FW) (type-a), the second one composed by a FW and Intrusion Detection System (IDS) (type-b); the third one composed by a FW, an IDS and a Network Address Translator (NAT) (type-c); the fourth one composed by a FW, an IDS, a NAT and a Proxy (type-d). The resource requirements of any SFC is characterized by a variable bandwidth that will be specified later and it is expressed by the Eqs (14) and (15) reported in Subsections VII-A and VII-B respectively.

VNF instances can be instantiated in the cloud infrastructure to support these SFs. According to the today's software availability we assume that any VNFI instance is able to support a maximum traffic capacity and requires the allocation of a given number of cores. We report in Table 6 the maximum capacity and the number of cores to be allocated for the various types (FW, IDS, NAT, Proxy) of VNFIs.

TABLE 6. Maximum processing capacity and allocated number of cores to the VNFIs implementing FW, IDS, NAT and Proxy.

VNFI Software	Maximum Processing Capacity ($C^{pr,max}$)	Number of cores allocated (n^c)
FW	900 Mb/s	4
IDS	600 Mb/s	8
NAT	900 Mb/s	2
Proxy	900 Mb/s	4

Next we compare the results of the proposed LCBC/DCA heuristics to the ones of the optimization problem in Subsection VII-A in the case of a small network. The effectiveness of the strategy proposed based on the knowledge of the peak traffic with respect to a reactive solution is illustrated in Subsection VII-B. All of the reported results are achieved as an average of fifty values corresponding to as many simulations.

A. COMPARISON BETWEEN ILP AND LCBC/DCA HEURISTICS

We compare the results of the proposed heuristics to the ones of the optimization problem formulated in Section V.

We assume that N SFC requests are generated each one randomly chosen among the ones considered (type-a, type-b, type-c and type-d). The peak traffic values $b_i(j)$ ($i = 1, \cdots N; j = 0, 1, \cdots T-1$) offered by the i-*th* SFC in the j-*th* Stationary Interval (SI) are chosen according to a classic daily traffic profile [11]. The values $b_i(j)$ are chosen as follows: i) the peak traffic values $b_i(0)$ ($i = 1, \cdots N$) requested in the 0-*th* SI is the highest one among the ones of all of the SIs in the cycle-stationary interval; ii) the values $b_i(0)$ ($i = 1, \cdots N$) are chosen in the set [100Mbps, 150Mbps, 200Mbps, 250Mbps, 300Mbps] according to a Zipf distribution [50] that is to say the probability that a bandwidth of $(100 + 50 * i)$ Mbps is generated for an SFC is given by the value c/i with c a normalization parameter of value 0.438; iii) the remaining values $b_i(j)$ ($i = 1, \cdots N; j = 1, \cdots T-1$) are chosen modulating the corresponding maximum traffic value $b_i(0)$ ($i = 1, \cdots N$) for a scale factor and reproducing a classical daily traffic profile; for this reason the remaining values $b_i(j)$ ($i = 1, \cdots N; j = 1, \cdots T-1$) can be expressed as follows:

$$b_i(j) = \begin{cases} b_i(0)(1 - 2\frac{j}{T}(1-\sigma)) \\ \qquad i = 1, \cdots, N; j = 1, \cdots, \frac{T}{2} \\ b_i(0)(1 - 2\frac{T-j}{T}(1-\sigma)) \\ \qquad i = 1, \cdots, N; j = \frac{T}{2}+1, \cdots, T-1 \end{cases} \tag{14}$$

where σ is the scale factor for the SI in which the traffic is minimum.

A number of SFCs is generated so as to produce a total traffic in the 0-*th* SI equal to 8 Gbps. The traffic is cycle-stationary with T equal to 8 and each SI has a duration time T_s equal to 3 hour. The parameter value σ is chosen equal to 0.1; this choice leads to a minimum bandwidth requested by any SFC equal to 90% of the maximum one. The choice of $T = 8$ and $\sigma = 0.1$ is according to a classical daily traffic variation.

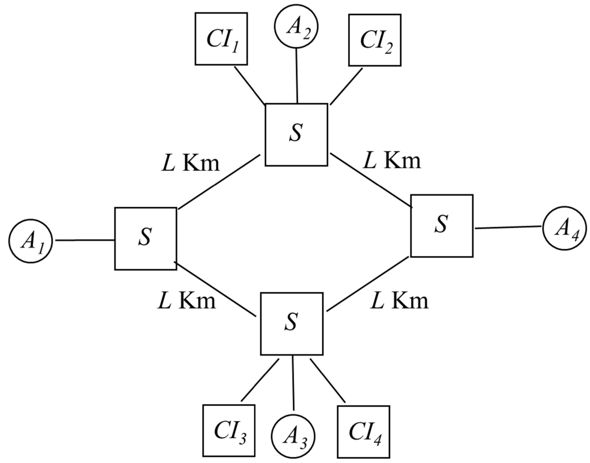


FIGURE 3. The network is composed by four switches, the access nodes A_1, A_2, A_3 and A_4 and the Cloud Infrastructures CI_1, CI_2, CI_3 and CI_4. Each CI is equipped with 48 core. The link bandwidths are chosen to be equal to 40 Gb/s.

Due to the high complexity of the optimization problem, we carry out the comparison in the case of the small network of Fig. 3 composed by four switches, four access nodes in which the SFCs are originated/terminated and four cloud infrastructures each one equipped with $N_{\bar{v}} = 48$ cores. We have chosen of performing the analysis in the case of CIs with only 48 cores to maintain low and acceptable the execution times of the ILP problem. The switches are interconnected by links of length L Km. The access nodes in which the SFCs are originated/terminated are randomly chosen among the access nodes of the network. The link bandwidths are chosen to be equal to 40 Gb/s.

From the Amazon rent price [23], [51] for a long term VM supporting a firewall we extrapolate core processing costs $c^{core}_{CI_1}$ and $c^{core}_{CI_2}$ equal to 5\$/h for Cloud Infrastructure CI_1 and CI_2 respectively. Conversely a value increased by the parameter α is chosen for the costs $c^{core}_{CI_3}$ and $c^{core}_{CI_4}$ of CI_3 and CI_4 respectively. Finally we assume a link bandwidth cost c_b equal to 10cent/GbKm.

We evaluate the VNFI graph by applying the procedure illustrated in [36] and revised in Appendix A. It applies an algorithm referred to as SFC Routing and Cloud and Bandwidth resource Allocation (SRCBA) algorithm. The input data of the SRCBA algorithm are the SFCs offered with the bandwidth requested in the 0-*th* SI that is the maximum traffic one. The output of the algorithm is a VNFI graph composed by 26 total nodes (4 access nodes and 22 VNFI nodes) and 86 virtual links. An initial embedding of the VNFI graph in the cloud and network infrastructure is also evaluated.

The comparison between the LCBC/DCA policy and the optimization results are reported in Table 7 where the total cost is reported for link lengths characterized by the parameter values L equal to 1 Km and 100 Km and for two cloud cost unbalancing factors α equal to 1.1 and 2. We also report

TABLE 7. Comparison of total costs for the LCBC/DCA algorithm and the output of the ILP.

A_o	10	20	30	40	50	60	70	80	90	100
ILP (α=1.1; L=1)	2853	6003	9054	12140	15410	17669	20651	22856	25029	27884
LCBC/DCA (α=1.1; L=1)	2889	6100	9416	12465	15697	18795	22280	25261	28726	32580
Error (%) (α=1.1; L=1)	1.25	1.59	3.85	2.60	1.83	5.99	7.31	9.52	12.87	14.41
ILP (α=1.1; L=100)	6057	11812	17860	22231	28130	32797	37469	43489	48026	51525
LCBC/DCA (α=1.1; L=100)	6168	12146	18960	24316	31222	37043	43550	51167	58279	65219
Error (%) (α=1.1; L=100)	1.81	2.75	5.80	8.57	9.90	11.46	13.96	15.01	17.59	21.00
ILP (α=2; L=1)	2838	6011	9180	12088	14928	17921	21805	24500	28419	33167
LCBC/DCA (α=2; L=1)	2888	6099	9405	12464	15702	18943	23509	27503	32544	38271
Error (%) (α=2; L=1)	1.74	1.44	2.40	3.01	4.93	5.40	7.25	10.92	12.68	13.34
ILP (α=2; L=100)	5659	12132	17071	22811	27094	34199	38715	46122	49331	56419
LCBC/DCA (α=2; L=100)	5805	12601	18018	24772	30429	38465	45023	54267	60814	71167
Error (%) (α=2; L=100)	2.52	3.72	5.26	7.92	10.96	11.09	14.01	15.01	18.88	20.72

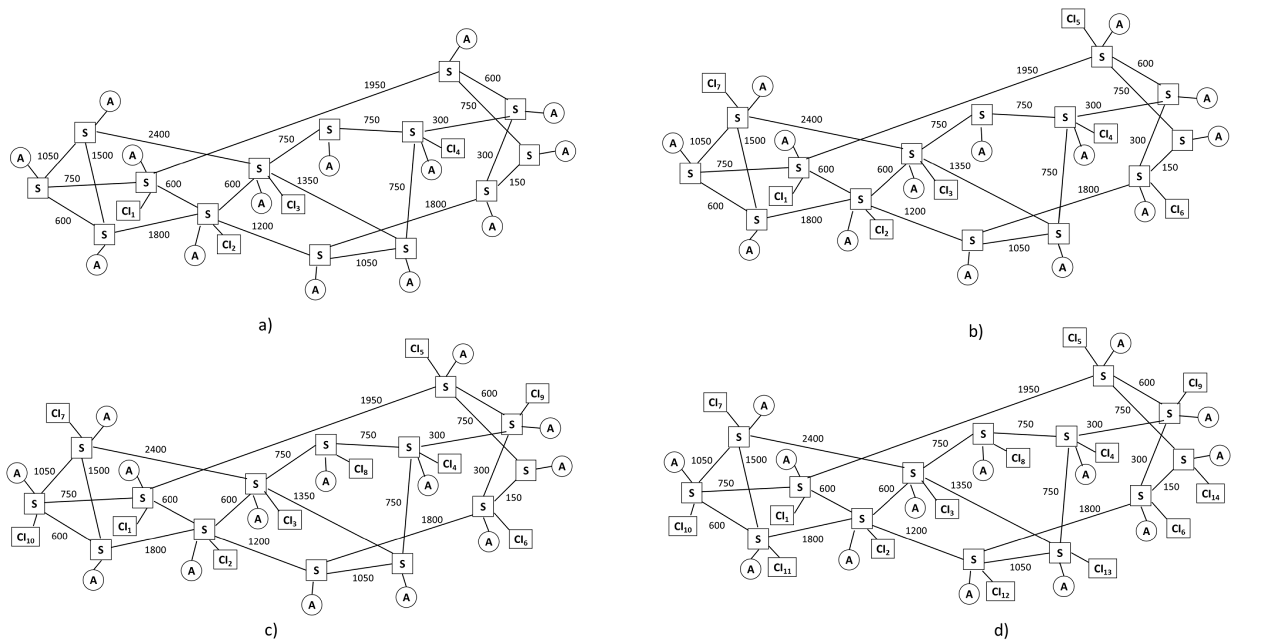


FIGURE 4. NSFNET network composed by 16 switches and 25 links. The four cases with a number N_{CI} of CIs equal to 4 (a), 7 (b), 10 (c) and 14 (d) are considered.

the percentage error on the total cost of the LCBC/DCA policy with respect to the ILP solution. We can notice how our LCBC/DCA heuristic allows us to achieve values near to the optimal ones. We can observe form Table 7 how the percentage error is always lower than 21%. The results, confirmed from other ones not reported in the paper for space reasons, proves the effectiveness of the LCBC/DCA policy especially for low values of L; in fact we have verified how the LCBC/DCA policy provides results near to the optimal ones in this low cost bandwidth scenario by trying as much as possible to migrate the VNFIs towards the least cost CI.

B. COST COMPARISON BETWEEN LCBC/DCA AND REACTIVE SOLUTIONS

We compare the proposed solution with the reactive solutions proposed in literature [18] in which the cloud and bandwidth resources are reconfigured when a traffic variation occurs; The reactive solution is based on the instantiation of on demand VNFIs according to the current traffic; these VNFIs are supported by short-term VMs whose cost is higher than the one of long-term VMs.

The results are reported in the case of the NSFNET network of Fig. 4 with 16 switches and 25 links. The four cases with a number N_{CI} of CIs equal to 4 (a), 7 (b), 10 (c) and 14 (d) are considered. For these four case studies the CIs are placed as illustrated in Figs 4.a, 4.b, 4.c and 4.d respectively. N SFCs, each one randomly chosen among the ones considered (type-a, type-b, type-c and type-d), are offered. The SFCs are originated and terminated in nodes of the network randomly chosen. A number of SFCs is generated so as to offer a maximum traffic A_o of 550 Gb/s. We assume a classical daily cycle-stationary traffic profile for the variation of bandwidth requested by any SFC; the duration time of the cycle-stationary period T_{cs} is assumed equal to 24 hours. We assume sinusoidal traffic variations occurring in time periods of duration T_m. In particular if we assume that T_{cs} is an integer multiple of T_m, we can express the traffic values $\xi_{i,s}$ for the i-*th* ($i = 1, \cdots, N$) SFC in the s-*th* ($s = 0, 1, \cdots T_{cs}/T_m - 1$) time period as follows:

$$\xi_{i,s} = \xi_i^p(1 + \cos(2\pi f_0 s T_m) + \omega)$$
$$i = 1, \cdots, N;\ s = 0, 1, \cdots T_{cs}/T_m - 1 \qquad (15)$$

where $f_0 = \frac{1}{T_{cs}}$ and we assume that the value ξ_i^p is chosen in the set [100Mbps, 150Mbps, 200Mbps, 250Mbps, 300Mbps] according to a Zipf distribution [50]. The parameter ω of expression (15) characterizes the minimum traffic value and it is chosen equal to 0.1.

The total cost of the Reactive Solution is evaluated as follows: i) according to the traffic values in the s-*th* time interval, the feasible embeddings and their cloud resource and bandwidth costs are evaluated by applying the LCBC algorithm; ii) the deployment costs involved for a change from the embedding applied in the (s-1)-*th* traffic interval to the ones determined to the point i) are evaluated; iii) the RS chooses among the evaluated embeddings, the one involving the minimum sum of the cloud resource, bandwidth and deployment costs.

Notice as the RS solution exploits the LCBC algorithm only to evaluate some a-priori embeddings of the VNFI graph into the cloud and network infrastructure; one of these embeddings is chosen according to the current traffic demand and by maximizing the difference between cloud and bandwidth cost saving and deployment cost involved. Conversely the proposed proactive procedure is based on both the LCBC and DCA algorithms and in particular it exploits the traffic prediction in the DCA algorithm by choosing the embeddings in the SIs so as to minimize the total cost in a cycle-stationary interval.

In this scenario we apply our LCBC/DCA solution in the case of $T = 8$ Stationary Intervals each one of time duration $T_s = 3$ hours. If we assume T_s to be an integer multiple of T_m the peak traffic $b_i(j)$ in the j-*th* ($j = 0, 1, \cdots, \frac{T_{cs}}{T_m} - 1$) SI of the i-*th* SFC ($i = 1, \cdots N$) is equal to $\max_{s \in [j\frac{T_{cs}}{T_m}, (j+1)\frac{T_{cs}}{T_m})} \xi_{i,s}$.

We provide in Fig. 5 a comparison between our LCBC/DCA solution based on peak traffic prediction and the Reactive Solution. The network scenario of Fig. 4.a is considered with the four Cloud Infrastructures CI_1, CI_2, CI_3 and CI_4. Each CI is equipped with $N_{\bar{v}} = 3072$ processing cores corresponding to 64 servers with 48 cores each one. Next we discuss the way in which we establish the cost of the cloud resources. For the long-term resource we follow the cost model of Amazon where such resources are referred to as Reserved Instances [51]. Reserved Instances are provided with a significant discount compared to On-Demand instance pricing (our short-term cloud resources), and can be purchased for a 1-year or 3-year term. Customers have the flexibility to change the availability zone, the instance size, and networking type of their Reserved Instances. The user can choose between three payment options when a Reserved Instance is purchased. With the All Upfront option, the user pays for the entire Reserved Instance term with one upfront payment. This option provides the user with the largest discount compared to On-Demand instance pricing. With the Partial Upfront option, the user makes a low upfront payment and are then charged a discounted hourly rate for the instance for the duration of the Reserved Instance term. The No Upfront option does not require any upfront payment and provides a discounted hourly rate for the duration of the term.

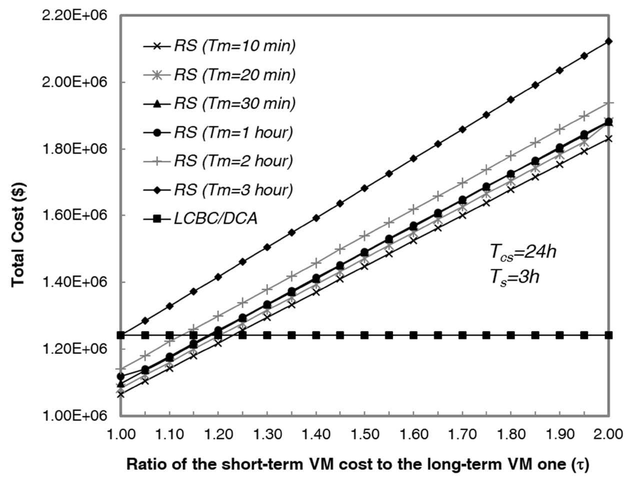


FIGURE 5. Comparison between the LCBC/DCA and Reactive Solutions in the case of the NSFNET network of Fig. 3. The total cost is reported as a function of the ratio τ of the short-term VM cost to the long-term VM one. T_m denotes the traffic variation time.

For our evaluation we choice the No Upfront option and report an hourly fee of the cloud resource cost. We establish our costs based on the one of the Amazon's $d2.8xlarge$ instance in the case for a 3-year term.

The core costs for the long-term VM are $c_{CI_1}^{core,LT} = c_{CI_2}^{core,LT} = 5$\$/h for the Cloud Infrastructures CI_1, CI_2 while the ones $c_{CI_3}^{core,LT}$ and $c_{CI_4}^{core,LT}$ are increased by 10% (α=1.1). The link bandwidth is fixed equal to 40Gb/s.

We report in Fig. 5 the total cost as a function of the ratio τ of the short-term core cost to the long-term core one. Then the short-term core cost $c_{CI_i}^{core,ST}$ is given by $\tau c_{CI_i}^{core,LT}$ ($i = 1 \cdots, 4$). The link bandwidth cost c_b is chosen to be equal to 10 $cent/GbKm$ while the deployment cost c_d equals 1\$. We report the values of the total costs for the RS by varying the values of T_m from 10 minutes to 3 hours. The decreases in T_m leads to assume more time-varying traffic profiles and the need to apply more frequently the Reactive Solution. One curve only is reported for the LCBC/DCA solution because the long-term VM cost as well as both the time duration and the peak traffic of the SI are held constant. From Fig. 5 we can observe how the application of RS allows for lower total cost values in the case of more time varying traffic profiles (lower T_m) and that is due to the possibility of RS in reconfiguring the cloud and bandwidth resources according to the current traffic demand. The high reconfiguration frequency requested to RS in highly variable traffic scenarios could lead to negative effects characterized by the information loss during the reconfiguration periods; the study of this effects is out of the scope of this paper and it has already been investigated in a previous paper of ours. [11] We can also observe from Fig. 5 that the total cost of the LCBC/DCA solution equals $1.24M\$$. For low values of τ, the LCBC/DCA policy has a total cost higher than RS solution due to its inefficient resource allocation based on the peak traffic. However as soon as the ratio τ reaches the value 1.25, the lowering of cloud resource cost of LCBC/DCA, due to the reservation in advance, leads to lower values of total cost.

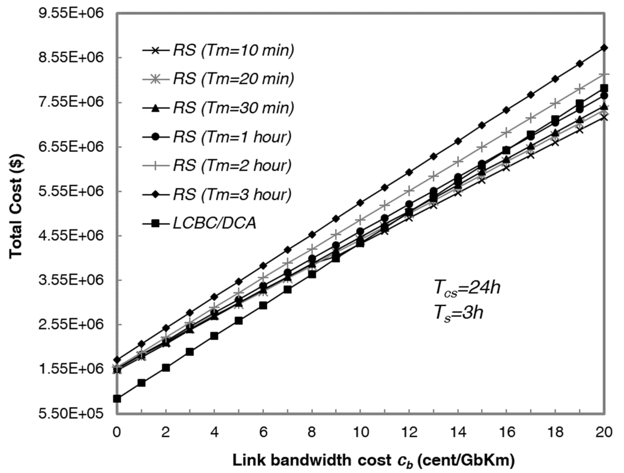


FIGURE 6. Comparison between the LCBC/DCA and reactive solutions in the case of the NSFNET network of Fig. 3. The total cost is reported as a function of the link bandwidth cost $c_b(cent/GbKm)$. The ratio τ of the short-term VM cost to the long-term VM one is chosen equal to 2.

As mentioned in [51] the cost saving of Reserved Instance (long-term cloud resource) with respect to the On Demand Instance (short-term cloud resource) is 50% on average that corresponds to a value of τ equal to 2. In this case, as illustrated in Fig. 5 the use of long-term resource can lead to a total cost saving equal to 34% with respect to the case in which short-term resources are used and T_m equals 1 hour. We report in Fig. 6 the comparison between LCBC/DCA solution and the RS in terms of the total cost when the link bandwidth cost c_b is varied from 0 to $20cent/GbKm$. We carry out the comparison in the case of $\tau = 1.4$ and when the same parameter values of the case study of Fig. 5 are chosen. We can observe how the LCBC/DCA solution, for c_b increasing, worse its total cost with respect to the RS. As a matter of example we notice how for values of c_b larger than or equal to 9 $cent/GbKm$ the LCBC/DCA solution has performance worse than RS for values of T_m from 10 minutes to 1 hour. This performance degradation is due to the higher bandwidth cost of LCBC/DCA with respect to RS that negatively impact the total cost when the link bandwidth cost c_b increases. In particular we report in Fig. 7 the bandwidth cost of the LCBC/DCA solution and RS as a function of the link bandwidth cost c_b. As expected, RS allows for lower bandwidth costs than LCBC/DCA solution. That is due to its capacity of reconfiguring the bandwidth resources when traffic variations occur.

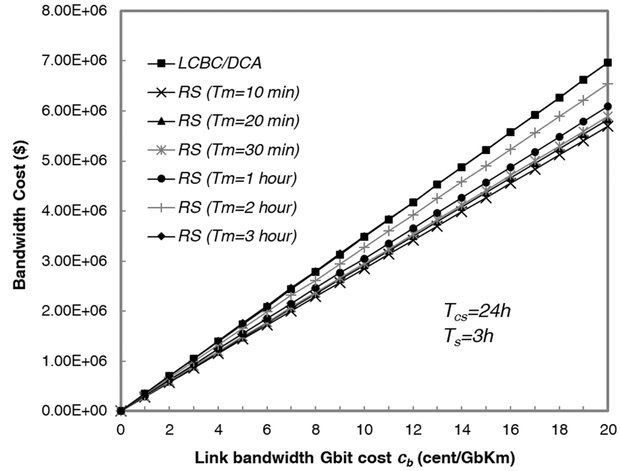


FIGURE 7. Bandwidth cost for the LCBC/DCA and reactive solutions as a function of the bandwidth cost $c_b(cent/GbKm)$. The ratio τ of the short-term VM cost to the long-term VM one is chosen equal to 2.

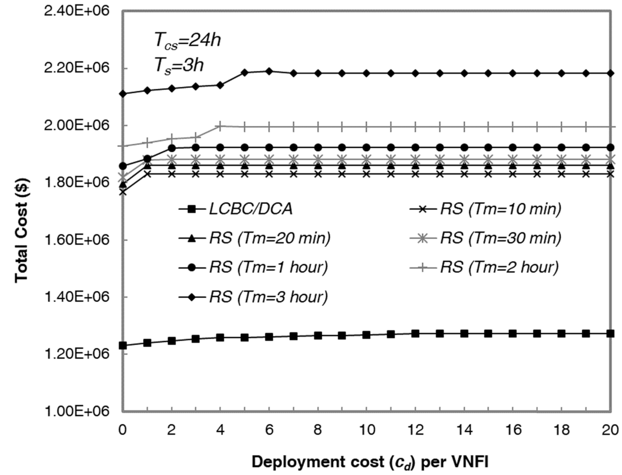


FIGURE 8. Total cost for the LCBC/DCA and reactive solutions as a function of the deployment cost c_d per VNFI. The ratio τ of the short-term VM cost to the long-term VM one is chosen equal to 1.4. The remaining parameter values are chosen as in the case study of Fig. 5.

Finally we study the impact of the deployment costs on the total cost for both LCBC/DCA and RS. We report in Fig. 8 the total cost as a function of the deployment cost c_d per VNFI charged by an InP when a VNFI is initialized in a Cloud Infrastructure. Even in this case we assume the deployment cost independent of the type of service function supported by the VNFI. The value of $\tau = 2$ is chosen while the remaining parameters are set up as in the case study of Fig. 5. Though c_d is increased up to the unrealistic value of 20\$, the deployment cost per VNFI slightly impacts the total cost for both LCBC/DCA and the RS. Better total costs are achieved for the LCBC/DCA policy. We report the total deployment cost in Fig. 9 as a function of the deployment cost c_d per VNFI. The values of c_d are always low for the LCBC/DCA solution because the network reconfigurations are performed at low frequency that is every $T_s = 3h$. From Fig. 9 we notice that the deployment cost drops to zero for c_d higher than or

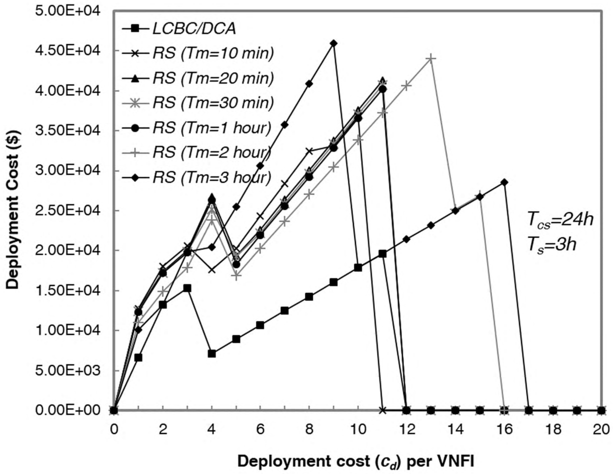


FIGURE 9. Deployment cost for the LCBC/DCA and reactive solutions as a function of the deployment cost c_d per VNFI. The ratio τ of the short-term VM cost to the long-term VM one is chosen equal to 1.4. The remaining parameter values are chosen as in the case study of Fig. 5.

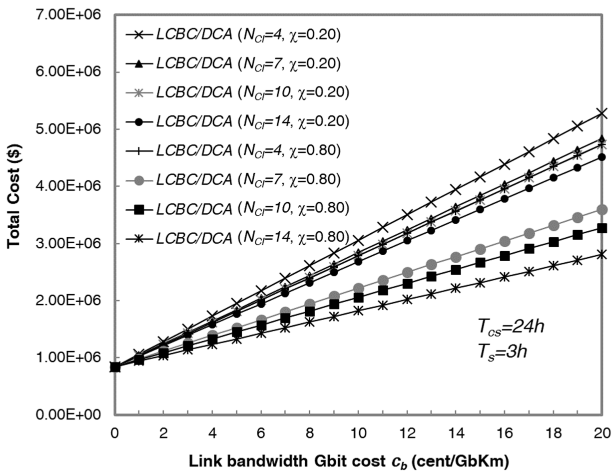


FIGURE 10. The total cost of the LCBC/DCA algorithm is reported as a function of the link bandwidth cost $c_b(cent/GbKm)$. The number N_{CI} of CIs is varied from 4 to 14 and the CIs are located as shown in Fig. 4. The core cost $c_{\bar{v}}^{core}$ is fixed equal to 5$/h. The fraction χ of local SFC requests is chosen equal to 0.2 and 0.8.

equal to 17$. The dropping to zero of the deployment cost is due to the fact that when c_d becomes higher than or equal to 17, the migrations are not performed at all according to the operation mode of LCBC/DCA and RS. The two algorithms provide a right trade-off between advantages in terms of low cloud resources and bandwidth costs from a side and low deployment costs from the other side.

Next we study the impact of the number N_{CI} of Cloud Infrastructure on the costs. We report in Fig. 10 the total cost as a function of the link bandwidth cost c_b and when the number N_{CI} is chosen equal to 4, 7, 10 and 14. The CIs are placed as indicated in Figs 4.a, 4.b, 4.c and 4.d respectively. We assume the same core cost $c_{\bar{v}}^{core} = 5\$/\text{h}$ for all of the CIs. The reported curves are parameterized versus the parameter χ that represents the fraction of local SFC requests that is the ones generated and terminated in a same access node. From Fig. 10 we can observe a decrease in total cost when the number N_{CI} of CIs increases especially when both the link bandwidth cost c_b and the fraction χ of local SFC requests are high. This decrease is consequence of the reduction in bandwidth cost due to the possibility of executing the Service Functions of any SFCs in CIs near the ingress/egress nodes in which the SFC is generated. As expected higher cost reduction is achieved for a higher fraction of local SFC requests. As a matter of example in the case $c_b = 18$ cent/GbKm and $\chi = 0.80$ the total cost equal 4346 M$ and 2612 M$ for N_{CI} equal to 4 and 14 respectively; in this case the use of all the CIs leads to a cost decrease by 40%. The bandwidth saving are highlighted in Fig. 11 where we report the bandwidth cost as a function of the link bandwidth cost c_b. As you can observe, Fig. 11 confirms the reduction in bandwidth cost when the number N_{CI} of CIs and the fraction χ of local SFC requests are increased.

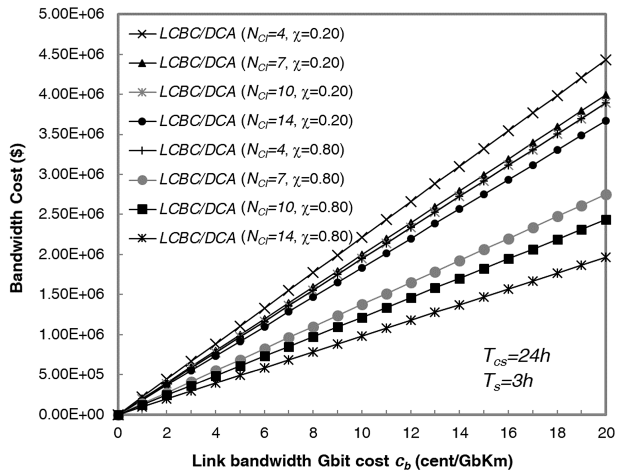


FIGURE 11. The bandwidth cost of the LCBC/DCA algorithm is reported as a function of the link bandwidth cost $c_b(cent/GbKm)$. The number N_{CI} of CIs is varied from 4 to 14 and the CIs are located as shown in Fig. 4. The core cost $c_{\bar{v}}^{core}$ is fixed equal to 5$/h. The fraction χ of local SFC requests is chosen equal to 0.2 and 0.8.

TABLE 8. Running time of the LCBC/DCA algorithm in the case of its application in a NSFNET network with number N_{CI} of CIs equal to 4, 7, 10 and 14 and according to the configuration of Fig. 4. The running times are expressed in seconds.

N_{CI}	**LCBC/DCA running time (s)**
4	48.1
7	209.3
10	435.1
14	660.7

Next we investigate the running time of the LCBC/DCA algorithm. It is reported in Table 8 when the number N_{CI} of CIs equals 4, 7, 10 and 14 and the LCBC/DCA algorithm is applied when the CIs are located as in Fig. 4. From the reported time values in Table 8 we can observe that the running time range from tens to thousands of seconds where obviously the highest times are achieved when all of the nodes are equipped with a CI ($N_{CI} = 14$). These configuration times highlight how the LCBC/DCA algorithm is able to quickly reconfigure the cloud and bandwidth resources when long-term traffic variations occur.

Finally we observe that the RS operation mode is based on the evaluation of pre-evaluated embeddings of the VNFI graph and the choice of one of these embeddings according to the current traffic condition; for this reason we can affirm that RS running time are in the order of few milliseconds that are not critical values for RS even when its application rate is high.

VIII. CONCLUSIONS

Network Function Virtualization is a key technology in 5G mobile networks both for the implementation of access functions and core functions. In 5G networks functionalities and nodes will be virtualized and executed in cloud infrastructures whose processing and memory resources will be managed by different Infrastructure Providers. The objective of this paper is to propose and investigate a solution for Multi-Provider NFV environments that according to the knowledge of the peak traffic of the SFC during stationary intervals allows for a reduction of cloud resource cost due to the possibility of renting in advance long-term Virtual Machines. We have given an ILP formulation of the problem of allocating cloud and bandwidth resources and by taking into account the deployment costs involved when the execution of a VNFI is moved in a different Cloud Infrastructure. Due to the NP-Hard complexity of the optimal problem, we have proposed the LCBC/DCA heuristic. By solving the optimization problem with a CPLEX solver, we have verified how the LCBC/DCA solution provides good solutions, near to the optimal ones. Finally we have compared the LCBC/DCA solution to a Reactive Solution that dependent on the traffic variation, asks for more frequent unpredictable reconfigurations with the needed of using on demand short-term Virtual Machines and without the possibility of renting in advance the cloud resources. We have shown how the LCBC/DCA solution in the case of low bandwidth cost, allows for lower total costs with respect to a Reactive Solution. Only when the bandwidth cost is prevalent, the RS outperforms the LCBC/DCA solution thanks to the possibility to allocate the bandwidth according to the current traffic and not to the peak ones as LCBC/DCA does.

APPENDIX A
SFC ROUTING AND CLOUD AND BANDWIDTH RESOURCE ALLOCATION (SRCBA) ALGORITHM

The SFC Routing and Cloud and Bandwidth resource Allocation (SRCBA) algorithm is based on the following two procedures. In the first one the N SFCs are sorted in decreasing bandwidth order. The second procedure allows for: i) the determination of the VNFI graph; ii) the determination of the CI in which the VNFIs are located; iii) the determination of the VNFIs in which the SFs are executed; iv) the determination of the network paths used to interconnect the VNFIs.

Next we give further clarification about the second procedure. It allows for the choice of CIs and network paths so as to minimize the sum of the cloud and bandwidth resource cost; it is applied for each SFC, it is devoted to build a multi-stage graph and to evaluate on it a least cost path that identifies the CIs on which the SFs of the SFCs have to executed either by using a VNFI already activated or by activating one new VNFI. The construction of the multi-stage graph G_i^{MS} for the i-*th* SFC is accomplished as follows. If M_i is the number of SFs in the i-*th* SFC, G_i^{MS} is composed by $M_i + 2$ stages, numbered from 0 to $M_i + 1$. The 0-*th* and the $(M_i + 1)$-*th* stages are composed by one node each one; a least cost path will be evaluated between these two nodes once the graph is built. The j-th stage contains n_j nodes, whose the generic one is named $v_{j,k}$ ($j = 1, \cdots, M_i, k = 1, \cdots, n_j$) where n_j is the number of nodes in each stage and it equals the number of CIs in which processing resources are available to execute the j-*th* SF of the i-*th* SFC. In particular notice how the execution of the SF may occur on a VNFI already activated or involve the instantiation of a new VNFI. The node $v_{j,k}$ of the multi-stage graph is characterized by a cost referred to as $c_{v_{j,k}}$ and characterizing the cost of executing the j-*th* SF of the i-*th* SFC in the CI associated to the node $v_{j,k}$. The cost depends whether a new VNFI needs to be activated and in this case it depends on the core cost c^{core}. We assume cost equal to zero in the case in which a VNFI can be re-used, otherwise the cost equals $c^{core}n_h^c$ if the SF is of h-*th* type. An edge of the multi-stage graph characterizes the possibility of interconnecting the two VNFIs located in the CIs associated to the two nodes of the edge. The edge is labeled with the average bandwidth cost involved in carrying the bandwidth b_i on the K-Shortest Paths connecting the CIs associated to the two nodes of the edge. The bandwidth allocation occurs on one out the K-Shortest Paths.

The computational complexity of the SRCBA heuristic depends on the evaluation procedure of the shortest path in the multi-stage graph. It is easy to prove that it is given by $C_{SO} = O(NM_{max}N_{CI}^2 log(M_{max}N_{CI}))$ where N is the number of offered SFCs, M_{max} is the number of SFs in the offered longest SFC and N_{CI} denotes the number of CIs.

REFERENCES

[1] F. Z. Yousaf, M. Bredel, S. Schaller, and F. Schneider, "NFV and SDN—Key technology enablers for 5G networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2468–2478, Nov. 2017.

[2] A. Leivadeas, G. Kesidis, M. Falkner, and I. Lambadaris, "A graph partitioning game theoretical approach for the VNF service chaining problem," *IEEE Trans. Netw. Service Manag.*, vol. 14, no. 4, pp. 890–903, Dec. 2017.

[3] M. Chiosi *et al.* (2012). *Network Functions Virtualization: An Introduction, Benefits, Enablers, Challenges and Call for Action*. [Online]. Available: https://portal.etsi.org/nfv/nfv_white_paper.pdf

[4] ETSI Industry Specification Group (ISG) NFV. (2015). *ETSI Group Specifications on Network Function Virtualization*. [Online]. Available: http://docbox.etsi.org/ISG/NFV/Open/Published/

[5] 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./Dec. 2014.

[6] J. Halpern and C. Pignataro, *Service Function Chaining (SFC) Architecture*, document RFC 7665, Oct. 2015. [Online]. Available: https://datatracker.ietf.org/doc/rfc7665/

[7] M. Yoshida, W. Shen, T. Kawabata, K. Minato, and W. Imajuku, "MORSA: A multi-objective resource scheduling algorithm for NFV infrastructure," in *Proc. 16th Asia–Pacific Netw. Oper. Manage. Symp.* Hsinchu, Taiwan: National Chiao Tung University Guangfu Campus, Sep. 2014, pp. 1–6.

[8] J. G. Herrera and J. F. Botero, "Resource allocation in NFV: A comprehensive survey," *IEEE Trans. Netw. Service Manag.*, vol. 13, no. 3, pp. 518–532, Sep. 2016.

[9] V. Eramo, A. Tosti, and E. Miucci, "Server resource dimensioning and routing of service function chain in NFV network architectures," *J. Elect. Comput. Eng.*, vol. 12, Aug. 2016, Art. no. 7139852.

[10] Y. Sang, B. Ji, R. Gagan Gupta, X. Du, and L. Ye. (2018). "Provably efficient algorithms for joint placement and allocation of virtual network functions." [Online]. Available: https://arxiv.org/pdf/1702.01154.pdf

[11] V. Eramo, E. Miucci, M. Ammar, and F. G. Lavacca, "An approach for service function chain routing and virtual function network instance migration in network function virtualization architectures," *IEEE/ACM Trans. Netw.*, vol. 25, no. 4, pp. 2008–2025, Aug. 2017. doi: 10.1109/TNET.2017.2668470.

[12] V. Eramo, M. Ammar, and F. G. Lavacca, "Migration energy aware reconfigurations of virtual network function instances in NFV architectures," *IEEE Access*, vol. 5, pp. 4927–4938, 2017. doi: 10.1109/ACCESS.2017.2685437.

[13] B. Kar, E. H.-K. Wu, and Y.-D. Lin, "Energy cost optimization in dynamic placement of Virtualized network function chains," *IEEE Trans. Netw. Service Manage.*, vol. 15, no. 1, pp. 372–386, Mar. 2018.

[14] F. Bari, S. R. Chowdhury, R. Ahmed, R. Boutaba, and O. C. M. B. Duarte, "Orchestrating virtualized network functions," *IEEE Trans. Netw. Service Manag.*, vol. 13, no. 4, pp. 725–739, Dec. 2016.

[15] W. Racheg, N. Ghrada, and M. F. Zhani, "Profit-driven resource provisioning in NFV-based environments," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Paris, France, May 2017, pp. 1–7.

[16] M. Obadia, J. Rougier, L. Iannone, V. Conan, and M. Brouet, "Revisiting NFV orchestration with routing games," in *Proc. IEEE Conf. Netw. Function Virtualization Softw. Defined Netw. (NFV-SDN)*, Palo Alto, CA, USA, Nov. 2016, pp. 107–113.

[17] F. Carpio, S. Dhahri, and A. Jukan, "VNF placement with replication for Loac balancing in NFV networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Paris, France, May 2017, pp. 1–6.

[18] M. Ghaznavi, A. Khan, N. Shahriar, K. Alsubhi, R. Ahmed, and R. Boutaba, "Elastic virtual network function placement," in *Proc. IEEE 4th Int. Conf. Cloud Netw. (CloudNet)*, Niagara Falls, ON, Canada, Oct. 2015, pp. 255–260.

[19] T. M. Nguyen, S. Fdida, and T. M. Pham, "A comprehensive resource management and placement for network function virtualization," in *Proc. IEEE Conf. Netw. Softwarization (NetSoft)*, Bologna, Italy, Aug. 2017, pp. 1–9.

[20] X. Wang, C. Wu, F. Le, A. Liu, Z. Li, and F. Lau, "Online VNF scaling in datacenters," in *Proc. IEEE 9th Int. Conf. Cloud Comput. (CLOUD)*, San Francisco, CA, USA, Jun. 2016, pp. 140–147.

[21] J. Liu, W. Lu, F. Zhou, P. Lu, and Z. Zhu, "On dynamic service function chain deployment and readjustment," *IEEE Trans. Netw. Service Manage.*, vol. 14, no. 3, pp. 543–553, Sep. 2017.

[22] V. Eramo, M. Ammar, and F. G. Lavacca, "Definition and evaluation of cold migration policies for the minimization of the energy consumption in NFV architectures," *Commun. Comput. Inf. Sci.*, vol. 18, pp. 45–60, Sep. 2017.

[23] X. Zhang, C. Wu, Z. Li, and F. C. M. Lau, "Proactive VNF provisioning with multi-timescale cloud resources: Fusing online learning and online optimization," in *Proc. IEEE Conf. Comput. Commun.*, Atlanta, GA, USA, May 2017, pp. 1–9.

[24] 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 Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 236–262, 1st Quart., 2016.

[25] W. Wang, B. Li, and B. Liang. (2013). "Reserve or not to reserve: Optimal online multi-instance acquisition in IaaS clouds." [Online]. Available: https://arxiv.org/abs/1305.5608

[26] L. Zheng, C. Joe-Wong, and C. G. Brinton, "On the viability of a cloud virtual service provider," in *Proc. ACM SIGMETRICS Int. Conf. Meas. Model. Comput. Sci.*, Jun. 2016, pp. 1–9.

[27] G. Maier, A. Feldmann, V. Paxson, and M. Allman, "On dominant characteristics of residential broadband Internet traffic," in *Proc. 9th ACM SIGCOMM Conf. Internet Meas.*, Chicago, IL, USA, Nov. 2009, pp. 90–102.

[28] H. Feng, J. Llorca, A. M. Tulino, D. Raz, and A. F. Molisch, "Approximation algorithms for the NFV service distribution problem," in *Proc. IEEE Conf. Comput. Commun.*, Atlanta, GA, USA, May 2017, pp. 1–4.

[29] Z. Liu, S. Wang, and Y. Wang. (2016). "Service function chaining resource allocation: A survey." [Online]. Available: https://arxiv.org/abs/1608.00095

[30] X. Li and C. Qian, "A survey of network function placement," in *Proc. 13th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Las Vegas, NV, USA, Jan. 2016, pp. 9–12.

[31] M. Ghaznavi, N. Shahriar, R. Ahmed, and R. Boutaba. (2018). "Service function chaining simplified." [Online]. Available: https://arxiv.org/abs/1601.00751

[32] H. Chen *et al.*, "MOSC: A method to assign the outsourcing of service function chain across multiple clouds," *Comput. Netw.*, vol. 133, pp. 166–182, Mar. 2018.

[33] B. Yi, X. Wang, S. K. Das, K. Li, and M. Huang, "A comprehensive survey of network function virtualization," *Comput. Netw.*, vol. 133, pp. 212–262, Mar. 2018.

[34] G. Sun, Y. Li, H. Yu, A. V. Vasilakos, X. Du, and M. Guizani, "Energy-efficient and traffic-aware service function chaining orchestration in multi-domain networks," *Future Gener. Comput. Syst.*, vol. 91, pp. 347–360, Feb. 2019.

[35] Y. Wang, P. Lu, W. Lu, and Z. Zhu, "Cost-efficient virtual network function graph (vNFG) provisioning in multidomain elastic optical networks," *J. Lightw. Technol.*, vol. 35, pp. 2712–2723, Jul. 1, 2017.

[36] V. Eramo and F. G. Lavacca, "Computing and bandwidth resource allocation in multi-provider NFV environment," *IEEE Commun. Lett.*, vol. 22, no. 10, pp. 2060–2063, Oct. 2018.

[37] X. Zhong, Y. Wang, X. Qiu, and S. Guo, "Cost-aware service function chain orchestration across multiple data centers," in *Proc. Netw. Oper. Manage. Symp.*, Taipei, Taiwan, Apr. 2018, pp. 23–27.

[38] V. Eramo and F. G. Lavacca, "Impact of the deployment costs on the cloud and bandwidth resource problems in multi-providers NFV environment," in *Proc. AEIT Conf.*, 2018, pp. 1–10.

[39] V. Eramo, E. Miucci, and M. Ammar, "Study of reconfiguration cost and energy aware VNE policies in cycle-stationary traffic scenarios," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1281–1297, Apr. 2016.

[40] R. Boutaba *et al.*, "A comprehensive survey on machine learning for networking: Evolution, applications and research opportunities," *J. Internet Services Appl.*, vol. 9, p. 16, Dec. 2018.

[41] M. Zubair Shafiq, L. Ji, A. X. Liu, and J. Wang, "Characterizing and modeling Internet traffic dynamics of cellular devices," in *Proc. 11th ACM SIGMETRICS Int. Conf. Meas. Modelling Comput. Syst.*, Jun. 2011, pp. 305–316.

[42] S. Gebert, R. Pries, D. Schlosser, and K. Heck, "Internet access traffic measurement and analysis," in *Proc. 4th Int. Conf. Traffic Monitor. Anal.*, Vienna, Austria, Aug. 2012, pp. 40–48.

[43] V. Eramo, E. Miucci, and M. Ammar, "Study of migration policies in energy-aware virtual router networks," *IEEE Commun. Lett.*, vol. 18, no. 11, pp. 1919–1922, Nov. 2014. doi: 10.1109/LCOMM.2014.2360190.

[44] C. Zeng, F. Liu, S. Chen, S. Jiang, and M. Li, "Demystifying the performance interference of co-located virtual network functions," in *Proc. IEEE Conf. Comput. Commun.*, Honolulu, HI, USA, Apr. 2018, pp. 16–19.

[45] X. Fei, F. Liu, H. Xu, and H. Jin, "Adaptive VNF scaling and flow routing with proactive demand prediction," in *Proc. IEEE Conf. Comput. Commun.*, Honolulu, HI, USA, Apr. 2018, pp. 16–19.

[46] V. Eramo, M. Listanti, F. G. Lavacca, P. Iovanna, G. Bottari, and F. Ponzini, "Trade-off between power and bandwidth consumption in a reconfigurable Xhaul network architecture," *IEEE Access*, vol. 4, pp. 9053–9065, 2016.

[47] S. G. Kulkarni *et al.*, "NFVnice: Dynamic backpressure and scheduling for NFV service chains," in *Proc. Conf. ACM Special Interest Group Data Commun.*, Aug. 2017, pp. 241–256.

[48] M. Garey and R. Graham, "Resource constrained scheduling as generalized bin packing," *J. Combinat. Theory, A*, vol. 21, no. 2, pp. 257–298, Nov. 1996.

[49] V. Eramo, M. Listanti, N. Caione, I. Russo, and G. Gasparro, "Optimization in the shortest path first computation for the routing software GNU zebra," *IEICE Trans. Commun.*, vol. E88-B, pp. 2644–2649, Jun. 2007.

[50] H. Kikuchi and K. Takahashi, "Zipf distribution model for quantifying risk of re-identification from trajectory data," in *Proc. 13th Annu. Conf. Privacy, Secur. Trust (PST)*, Jul. 2015, pp. 14–21.

[51] *EC2—Amazon Web Services*. [Online]. Available: https://aws.amazon.com/ec2/

VINCENZO ERAMO received the Laurea degree in electronics engineering and the Dottorato di Ricerca (Ph.D.) degree in information and communications engineering from the Sapienza University of Rome, in 1995 and 2001, respectively. In 1996, he was a Researcher with the Scuola Superiore Reiss Romoli. In 1997, he joined the Telecommunication Network Planning Group, Fondazione Ugo Bordoni, as a Researcher. From 2002 to 2005, he was an Assistant Professor and from 2006 to 2010, he was an Aggregate Professor with the INFOCOM Department, Sapienza University of Rome, where he is currently an Associate Professor with the Department of Engineering of Information, Electronics and Telecommunications. His research activities have been carried out in the framework of national and international projects. In particular, he was a Scientific Coordinator for the Sapienza University of Rome of E-PhotoONe+ and BONE, two Networks of Excellence focusing on the study of optical networks and financed by European Commissions (FP6 and FP7), in 2006–2007 and 2008–2011, respectively. He was an Associate Editor of the IEEE TRANSACTIONS ON COMPUTER, from 2011 to 2015, and he has been an Associate Editor of the IEEE COMMUNICATION LETTERS, since 2014. He received the Exemplary Editor Award from the IEEE COMMUNICATIONS LETTER, in 2016 and 2017.

FRANCESCO GIACINTO LAVACCA received the Laurea (M.Sc.) degree *(cum laude)* in electronic engineering and the Ph.D. degree in information technology from the Sapienza University of Rome, Italy, in 2013 and 2017, respectively. From 2017 to 2018, he was a Postdoctoral Researcher with the Department of Information, Electronics and Telecommunication Engineering (DIET). He has been a Visiting Researcher with the College of Computing, Georgia Institute of Technology, Atlanta, GA, USA. Furthermore, he was involved in the framework of national and international projects, like Advanced Avionic Architecture (AAA) and Nano Micro Launch Vehicle (NMLV) with Italian Space Agency (ASI) and European Space Agency (ESA), respectively. His current research interests include all-optical networks and switching architectures, 5G networks, network function virtualization, time-triggered, and deterministic Ethernet.