Decentralized Scalable Dynamic Load Balancing among Virtual Network Slice Instantiations

Michele Aicardi[†], Roberto Bruschi[*], Franco Davoli[*‡], Paolo Lago[□] and Jane Frances Pajo[*‡]
[†]*DIBRIS – University of Genoa*, Genoa, Italy | [‡]*DITEN – University of Genoa*, Genoa, Italy
[*]*CNIT S3ITI National Laboratory*, Genoa, Italy | [□]*NPE S.r.l. - De' Longhi Group*, Longarone, Italy
michele.aicardi@unige.it | roberto.bruschi@cnit.it | franco.davoli@unige.it | paolo.lago85@gmail.com | jane.pajo@tnt-lab.unige.it

***Abstract*— In the virtualized environment of 5G networks, the control and management of dynamic network slices poses a set of challenges that are still largely unsolved. Though the architectural framework and the elements of abstraction and orchestration mechanisms have been defined, the dynamic orchestration of resources based on them entails the adoption of existing sophisticated control techniques, or the design of new ones for the specific environment. In the present paper, we address the problem of load balancing among multiple network service chains (which represent network slice instantiations of a Network Service Provider referring to a specific vertical application) originating from different Points of Presence (PoPs). For scalability reasons, we want to maintain the problem within an informationally decentralized setting, where each PoP has the knowledge of the aggregate workload generated by the slice users accessing through it, but not of that of the other PoPs (to avoid the exchange of information for control purposes). By taking also into account power consumption policies of the Infrastructure Provider, we find a set of candidate team-optimal solutions to this load-balancing problem, which are characterized by piecewise-linear functions, and compare their performance with that of other resource allocation strategies.**

Keywords*— *Network Functions Virtualization, Network Slices, Load Balancing, Network Energy Efficiency, Team Decision Theory.

I. INTRODUCTION

Within the challenges posed by the Future Internet in general, and particularly by the strong wireless/wired integration of the 5th generation wireless (5G) environment, four broad topics, among others, can be seen as interacting and mutually influencing: i) flexibility, programmability and virtualization of network functions and services, ii) performance requirements (in terms of users' Quality of Experience – QoE – and its mapping onto Quality of Service – QoS – in the network), iii) energy efficiency, and iv) network management and control.

The first item stems from the evolution of the network towards a multi-purpose "softwarized" service-aware platform upon a heterogeneous infrastructure [1], to deal with diverse and integrating paradigms as 5G, the Internet of Services, the Internet of Things (IoT), network-integrated cloud/fog computing services, just to quote a few examples. Performance issues have to deal with the very strong requirements imposed by 5G Key Performance Indicators (KPIs) [2] and energy-awareness cannot be neglected in view of sustainability, environmental concerns, and operational costs. In this scenario, network management and control strategies are essential to orchestrate all needed functionalities, supervise and optimize the allocation of resources, to ensure that KPIs are met for network slices [3]-[5] under the dynamic evolution of user-generated traffic, multiple tenants, service and infrastructure providers. Indeed, though a general reduction in Operational Expenditures (OpEx) is expected [6] (besides the reduction in Capital Expenditures – CapEx – entailed by the use of general-purpose hardware (HW)) from the upcoming revolution in networking paradigms brought forth by Software Defined Networking (SDN) [7] and Network Functions Virtualization (NFV) [8], this reduction will not come without the adoption of specific management and control solutions.

As regards in particular energy efficiency, the massive introduction of general-purpose HW enabled by NFV would tend *per se* to increase power requests with respect to specialised HW solutions [9], in the absence of specific control actions. Among the various techniques that can be adopted to this purpose to implement Control Policies (CPs) in network processing devices, Dynamic Adaptation ones consist of modulating the processing rate (Adaptive Rate – AR) or of exploiting low power consumption states in idle periods (Low Power Idle – LPI) [10]. In virtualized networks, where a collection of network service chains must be allocated on physical network nodes, the latter may apply Local Control Policies (LCPs) implementing such dynamic adaptation concepts. In more detail, one such chain is a set of one or more Virtual Network Functions (VNFs) grouped together to provide specific service functionality [11] in a VNF-Forwarding Graph (VNF-FG) – an oriented graph, where each node corresponds to a particular VNF and each edge describes the operational flow exchanged between a pair of VNFs. A network service request can be allocated on dedicated HW or by using resources deployed by an Edge- or Fog-Computing Provider, residing in a datacentre that processes the request through virtualized instances.

Focusing on the latter type of service deployment, in a previous paper [12] we have introduced a Game-Theory-based solution for energy-aware allocation of VNFs and determined the existence of a Nash Equilibrium. In this paper, we assume a different point of view, whereby the actors of the game are Decision Makers (DMs) located at the Points of Presence (PoPs) of a Network Service Provider (NSP), through which user applications access a specific Network Service offered through a VNF-FG by the NSP, which provides a network slice [9] tailored for a given vertical application. The NSP is in its turn a specific tenant of the Infrastructure Provider (IPr, which is the owner of the physical machines) and runs the VNFs on Virtual Machines (VMs) deployed on the IPr's HW. The DMs are meant to balance the workload offered to their own PoPs among a number of VNF-FGs performing the required functionality; VNF-FGs are differentiated in terms of a global cost that accounts for both performance and energy consumption. For scalability, signalling reduction and fast reactivity reasons, we are interested in finding informationally decentralized (per-PoP)

load balancing solutions in *strategic* form (i.e., mapping available instantaneous information into the required decisions, over the whole possible range of observable variables).

The remainder of the paper is organized as follows. In the next Section, we briefly describe virtualized network services, in the light of the NFV architectural framework represented in ESTI standards (see, among others, [11], [13]), and offer a comprehensive summary on the approach and main design choices at the foundations of the proposed load balancing mechanism. In Section III, we introduce a cost function that accounts for energy and performance aspects. Section IV is devoted to the VNF-FG Allocation Problem (VNF-FG-AP) in the presence of multiple VMs per network service. Extensive numerical examples are presented in Section V, while Section VI contains the conclusions.

II. Anatomy of Virtualized Network Services

A. The Architectural Perspective

The ETSI NFV standard [11] defined an architectural framework where, through the adoption of well-known and widespread Information Technology (IT) virtualization techniques, the operational domains of infrastructure and service providers (e.g., vertical industries, over-the-top network providers, etc.) are fully split. In detail, service providers can define their own network services and instantiate their components over infrastructure resources (in terms of network, computing and storage) acquired *as-a-Service* from different IPrs. At the same time, IPrs can simultaneously host multiple service providers in the same NFV PoPs' datacenters.

From a "cloud" perspective, the NFV framework can be defined as a *multi-domain* and *multi-tenant* architecture. *Service Providers* are envisioned to compose and orchestrate the lifecycle of their NFV services, instantiating their components (i.e., VMs) over the resources acquired as-a-Service from PoPs. *Infrastructure Providers* are expected to handle and keep running service components in their infrastructure in an efficient fashion (e.g., by consolidating VMs/execution containers in a subset of servers, in order to reduce the energy consumption of PoPs, and/or by applying energy-aware LCPs on their devices).

The ETSI NFV working group defines a Network Service as a graph of network functions (either virtual or physical) connecting end-points (i.e., specific network terminations). In its turn, each network function can be hierarchically composed by further functions or by components. VNF components represent the lowest decomposition level and might correspond to VMs or other kinds of execution containers.

In addition to the previous ETSI definitions, 3GPP and NGMN recently introduced the *network slicing* concept into 5G ecosystem specifications [3][4]. A network slice can be roughly summarized as a virtual projection of a 5G network with all the functionalities, isolation level, and capabilities customized according to the needs of the vertical applications. A network slice is defined to be composed of one or more interconnected logical subnetworks which might provide different functionalities (e.g., radio access, packet core, etc.) or that can be even shared with other network slices. It is reasonable to assume that a slice subnetwork corresponds to one or more NFV services, maintained and orchestrated by the ETSI MANO orchestrator.

B. The Performance Perspective

One of the main objectives of the Orchestrator defined by ETSI NFV MANO (NFV Management and Orchestration) [13], along with the help of VNF managers, is to automatically and dynamically manage the service graph and the configuration of any single component to cope with the offered load and performance requirements. This aspect is directly inherited from the *elasticity* capability in today's cloud computing technologies [14], which allows tenants to dynamically acquire and release resources *as-a-Service* depending on their needs. In this respect, two base techniques, namely *vertical* and *horizontal scaling* can be used in a non-exclusive fashion [15].

Anyway, as described in sub-section III-C in more detail, it is well known that the overall performance of the software running inside the execution container might not scale linearly with respect to the associated resources, depending on the parallelization degree that the algorithms (and the relative implementations) in the software can provide. Moreover, vertical scaling is clearly upper limited by the resources available in the hosting servers. Horizontal scaling (also referred to as "scaling up/down") removes the previous limitation, since it allows creating/removing copies of the same VM on-the-fly and balancing the load among these copies accordingly. However, given the "centralized" nature of cloud computing scenarios where these techniques have been originally applied, the load balancing process is usually designed to work with VMs residing in the same datacenter. Its application to multi-domain scenarios, as addressed by the NFV specifications (i.e., multiple PoPs), might lead to highly inefficient network configurations [16], where traffic may bounce between datacenters at any "load-balanced" VMs in the service chain.

From this observation, the need becomes evident of load-balancing techniques (perhaps closer to control-plane run-time decisions than to the MANO framework) able to consider the entire end-to-end chain deployment, and to optimally steer traffic where horizontally scaled copies of the same VM, or even other VMs implementing equivalent functionalities, can be exploited. A potential problem associated to such end-to-end load-balancing techniques is the quantity of information to be maintained and synchronized among all the distributed elements concurring to the service implementation. Distributed optimization strategies (capable of mapping local information into dynamic control actions) can reduce the quantity of signalling to be exchanged for control purposes and scale better as a viable solution for large-scale systems like the upcoming 5G networks.

C. Main Approach

We suppose a given number of VNF-FGs to be active to provide the networking functionalities requested by the different instantiations of the user's application service. VNFs composing the chains are deployed on a pool of computational resources, but each specific VNF is associated to an execution container or VM, or to a set of VMs performing the same function, made available upon the HW of the IPr (we will refer to VMs in the following). Upon request of the network service pertaining to the slice, the PoP DM has the possibility to choose the chain of VNFs that are needed by the service, among a number of possible alternatives. An overview of the considered scenario is illustrated in Fig. 1. Different VNF-FGs may require common VNFs, and therefore they may share the

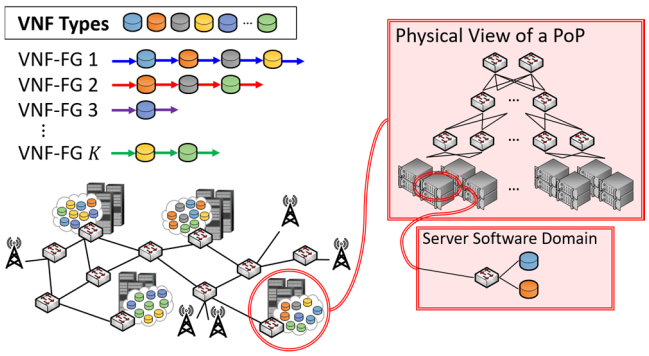


Fig. 1. Overview of the scenario.

computational capacity of the VMs performing them. Each chain is characterized by an execution cost, which is the sum of the costs pertaining to each VNF composing it. The cost of a specific VNF is determined by the workload on the VM where it is executed and by the energy-aware LCP adopted by the IPr. We suppose each PoP DM to be aware of the LCPs and of its own workload (the rate of service requests), but not of the workloads generated at the other PoPs. As already mentioned, this information decentralization constraint essentially stems from a scalability and signalling issue, as the NSP may have a number of different PoPs sharing the VNFs of different VNF-FGs, whose workload may change relatively frequently, depending on the user attachments being served (that may vary over an observation period owing also to users' mobility and performance requirements).

In this informationally decentralized setting, we want to determine the decentralized control policy that maps the possible workload values at the PoPs to the shares of this workload to be assigned to each of the VNF-FGs pertaining to the specific network service. The problem is posed in a team-theoretical setting ([17][18]), as all PoP DMs of the NSP have the same goal, which consists of the minimization of a common overall cost for the usage of the resources, but they possess different information on their respective incoming flows.

By adopting specific models for a server's core energy consumption [19] and for the power/delay cost of a server [20], we construct the cost associated to each VM as the product of the energy consumption and the processing delay. Under a certain behaviour of the IPr, this cost turns out to be quadratic in the total workload on the HW hosting the VM; this fact renders the team problem mathematically equivalent to the one we considered in [21] in a different setting and allows the application of the solution derived therein.

III. The Optimization Cost of NFV Services

Owing to the considerations in Sect. II, it is clear that NFV services' operations and deployment strategies have to be optimized by finely controlling the trade-off between the dimensioning of virtual resources acquired as-a-Service (vCPUs, RAM, network resources, etc.) and the performance levels (e.g., processing time) that VNFs instantiated on those resources can provide.

To capture this trade-off, we define the optimization cost J associated to the VMs implementing the complete slice functionality as the sum of the products of processing delays D_j and of the energy consumption Φ_j induced by every VM instantiated onto the NFV infrastructure:

$$J = \sum_{j=1}^{M} J_j = \sum_{j=1}^{M} D_j \, \Phi_j \tag{1}$$

where M is the number of VMs deployed by the tenant for the network slice. The product between the processing delay and the energy consumption has been widely used in recent years for modelling and optimizing the design of modern (virtualization-ready) computing systems and components, like many-core processors and systems-on-chip, according to the performance of software applications [20].

Given the advanced support/capabilities of recent computing platforms and virtualization hypervisors, VMs running on a server can be considered as almost completely isolated among themselves, and – especially in high performance scenarios as NFV – exploiting different server internal components (e.g., CPU cores, etc.). In detail, considering the j-th VM in the NFV service, we assume that c_j vCPUs will be bound to c_j physical CPU threads/cores available in the server by the virtualization hypervisor. Even though the overall VM processing capacity scales linearly according to c_j, software applications hosted in the VM are well-known to exhibit a different performance behavior which depends on their parallelizability.

In this respect, the well-known Amdahl's Law and its recent generalizations [20] suggest that the software-level performance depends on the number of available cores c_j and on their capacity μ_j by a speed-up factor S_j:

$$\mu_j(c_j) = S_j \, \mu_j = \frac{1}{\sum_{n=1}^{c_j} \frac{\beta_n^{(j)}}{n}} \mu_j \tag{2}$$

where $\beta_n^{(j)}$ is the n-th fractional component into which an algorithm implementing the j-th NFV service component can be split ($\sum_{n=1}^{c_j} \beta_n^{(j)} = 1$). In general, $\beta_n^{(j)}$ is parallelizable on n cores; $\beta_1^{(j)}$ represents the fraction that is not parallelizable and, without loss of generality, we number the cores in ascending order of utilization.

Now, we want to introduce a cost function capable of capturing both processing delay and energy aspects of VM j. In a very simple formulation, by considering the aggregate action of the speedup introduced by parallelization, the delay term can be taken as that of a M/M/1 queueing system; by indicating with f_j the total load on VM j,

$$D_j(c_j) = \frac{1}{S_j \mu_j - f_j} \tag{3}$$

Despite the possible inaccuracy of the M/M/1 model, its simple expression for $D_j(c_j)$ results to be an effective penalty function with respect to the saturation of the processor's capacity, which is an essential characteristic to be reflected in our optimization problem.

Regarding the power consumption, we start from the model considered in [19], which takes into account the presence of both frequency scaling and low power idle effects in the power profile of a single CPU core; namely, if δ_{jn} is the load fraction processed on its assigned core n, the power consumption Φ_{jn} is given by

$$\Phi_{jn} = K_j \mu_j^3 \left(\frac{\delta_{jn} \, f_j}{\mu_j} \right)^{1/\alpha_j} \tag{4}$$

where $K_j \in \mathbb{R}^+$ and $\mathbb{R}^+ \ni \alpha_j \geq 1$ are parameters depending on the HW type (we denote by $\mathbb{R}^+$ the set of positive real numbers).

Given our ordering, δ_{jn} turns out to be:

$$\delta_{jn} = \sum_{s=n}^{c_j} \frac{\beta_s}{s} \tag{5}$$

Thus, the power consumption induced by VM j on the c_j CPU cores can be expressed as follows:

$$\Phi_j = \sum_{n=1}^{c_j} \Phi_{jn} = \sum_{n=1}^{c_j} K_j \mu_j^3 \left(\frac{\delta_{jn} f_j}{\mu_j} \right)^{1/\alpha_j} \tag{6}$$

To each VM j we associate the cost J_j expressed as the product of processing delay and power consumption:

$$J_j = \Phi_j D_j = \left[\sum_{n=1}^{c_j} K_j \mu_j^3 \left(\frac{\delta_{jn} f_j}{\mu_j} \right)^{1/\alpha_j} \right] \cdot \frac{1}{S_j \mu_j - f_j}$$

$$= \frac{K_j \mu_j^{3-1/\alpha_j} f_j^{1/\alpha_j}}{S_j \mu_j - f_j} \left[\sum_{n=1}^{c_j} \left(\delta_{jn} \right)^{1/\alpha_j} \right] = K_j' \frac{f_j^{1/\alpha_j} \mu_j^{3-1/\alpha_j}}{S_j \mu_j - f_j} \tag{7}$$

In (7) above, we have collected all multiplicative coefficients related to VM j in the term K_j', which, given the characteristics of the specific network application and of the HW, is a known constant. By minimizing J_j with respect to μ_j we obtain:

$$\mu_j^* = \underset{\mu_j}{\operatorname{argmin}} J_j = \frac{3\alpha_j - 1}{S_j (2\alpha_j - 1)} f_j = \theta_j f_j \tag{8}$$

$$J_j^* = \min_{\mu_j} J_j = K_j' \frac{\theta_j^{3-1/\alpha_j}}{S_j \theta_j - 1} f_j^2 = h_j f_j^2 \tag{9}$$

having defined

$$\theta_j = (3\alpha_j - 1)/S_j(2\alpha_j - 1),\ h_j = K_j' \theta_j^{3-1/\alpha_j} / (S_j \theta_j - 1).$$

Thus, the application of a simple proportional control law like (8) on the part of the IPr has the effect of making the cost associated to the VM using its core(s) quadratic in the total load[1]. We will exploit this fact in the next Section to formulate our quadratic constrained team optimization problem.

IV. The Team Optimization Problem Statement and Solution

The total flow offered to the j-th VM implementing a specific VNF is composed in general by a number of contributions pertaining to the VNF-FGs that use its functionality. Let S be the total number of PoPs, F the total number of VNF-FGs, and $\mathcal{F}_i$, with $|\mathcal{F}_i| = F_i \leq F, i = 1, \dots, S$, the subset of VNF-FGs used by the i-th PoP. We indicate by u_{ki}, $k \in \mathcal{F}_i, i = 1, \dots, S$, the fraction of the workload r^i generated at PoP i that is offered to VNF-FG k. Let $\underline{r} = [r^1, \dots, r^S]^T$ (the superscript T indicates transpose) be the vector collecting all PoP workloads, and further let $\mathcal{V}_j$ be the set of VNF-FGs that use the services of VM j. We assume the components of $\underline{r}$ to be independent non-negative continuous random variables with a given probability distribution. We can then state the following team optimization problem.

$$\min_{\gamma_{ki}(\cdot),\ k \in \mathcal{F}_i;\ i=1,\dots,S} \bar{J} \tag{10}$$

where

$$\bar{J} = \frac{1}{2} \underset{\underline{r}}{E} \left\{ \sum_{j=1}^{M} w_j \left(\sum_{i=1}^{S} \sum_{k \in \mathcal{F}_i \cap \mathcal{V}_j} u_{ki} \right)^2 \right\} \tag{11}$$

with

$$u_{ki} = \gamma_{ki}(r^i)\ ,\ \ k \in \mathcal{F}_i; i = 1, \dots, S \tag{12}$$

under the constraints

$$\sum_{k \in \mathcal{F}_i} u_{ki} = r^i\ ,\quad i = 1, \dots, S \tag{13}$$

$$u_{ki} \geq 0\ ,\quad k \in \mathcal{F}_i; i = 1, \dots, S \tag{14}$$

In (11), w_j is a weighting coefficient that accounts for both the value of h_j (stemming from the underlying HW and the parallelizability of the VNF code) and the influence of the network topology on the contributing flows that enter the VM (as they may traverse different network paths to reach it from the previous VMs in their chains; e.g., the coefficient may be generated by a weighted sum of link costs that account for the "distance" of each source from VM j in the given network topology). Equations (12) entail a decentralization constraint. In other words, DM i decides on the shares of its workload among the VNF-FGs only on the basis of the knowledge of its own workload, and not of that of the others. The only centralized information is constituted by the a priori knowledge (the number of VMs and of DMs, the topology and the probability distributions of the inputs).

Following the same line of reasoning as in our previous work [21], we consider finding person-by-person optimal (p.b.p.o.) strategies [17][18] of the form (12) for the above problem. By defining $\gamma^i(\cdot) = \{\gamma_{ki}(\cdot), k \in \mathcal{F}_i\},\ i = 1, \dots, S$, the p.b.p.o. strategy of DM i, , $i = 1, \dots, S$, entails the minimization of the cost (11), under fixed (functional) values of the strategies of the other agents $\gamma^{-i}(\cdot) = \{\gamma_{kj}(\cdot), k \in \mathcal{F}_j; j = 1, \dots, S, j \neq i\}$; namely, we are looking for functions $\gamma^{i^*}(\cdot)$ such that

$$\gamma^{i^*}(\cdot) = \underset{\gamma^i(\cdot)}{\operatorname{argmin}} \bar{J}_i[\gamma^i(\cdot), \gamma^{-i^*}(\cdot)]$$

$$= \underset{\gamma^i(\cdot)}{\operatorname{argmin}} \frac{1}{2} \underset{\underline{r}}{E} \left\{ \sum_{j=1}^{M} w_j \left[\sum_{k \in \mathcal{F}_i \cap \mathcal{V}_j} \gamma_{ki}(r^i) + \sum_{\substack{\ell=1 \\ \ell \neq i}}^{S} \sum_{k \in \mathcal{F}_\ell \cap \mathcal{V}_j} \gamma_{k\ell}^*(r^\ell) \right]^2 \right\}$$

$$= \underset{u^i}{\operatorname{argmin}} \frac{1}{2} \underset{r^\ell, \ell=1,\dots,S, \ell \neq i}{E} \left\{ \sum_{j=1}^{M} w_j \left[\sum_{k \in \mathcal{F}_i \cap \mathcal{V}_j} u_{ki} \right. \right.$$

[1] As noted in [12], we are considering a continuous solution to the server operating capacity adjustment. In practice, the physical resources allow a discrete set of working frequencies, with corresponding processing capacities. This would also ensure that the processing speed does not decrease below a lower threshold, avoiding excessive delay in the case of low load.

$$+\sum_{\substack{\ell=1\\ \ell\neq i}}^{S}\sum_{k\in\mathcal{F}_\ell\cap\mathcal{V}_j}\gamma^*_{k\ell}(r^\ell)\Bigg]^2\Bigg|\, r^i\Bigg\},$$

$$i=1,\dots,S,\ \ \forall\, r^i \tag{15}$$

where we have defined $u^i=\left[u_{1i},\dots,u_{F_i i}\right]^T$. Conditioning the expectation in the third line in (15) transforms the functional optimization problem of DM i into an ordinary minimization, which can be handled by the application of Karush-Kuhn-Tucker conditions. However, the solution for DM i depends on the average of the strategies of the other DMs, as can be seen by the coefficient of the linear term that arises in (15) by expanding the square. By defining

$$(\eta_i^m)^*=\sum_{\substack{\ell=1\\ \ell\neq i}}^{S}\sum_{k\in\mathcal{F}_\ell\cap\mathcal{V}_m}\underset{r^\ell}{E}\{\gamma^*_{k\ell}(r^\ell)\},$$

$$m=1,\dots M, i=1,\dots,S \tag{16}$$

the quantities $(\eta_i^m)^*$ can be treated as a set of parameters characterizing the strategies. By indicating the parametrized p.b.p.o. strategies as $\hat{\gamma}^{i*}\left(r^i,\{(\eta_j^n)^*, n=1,\dots,M; j=1,\dots,S, j\neq i\}\right)$, the unknown quantities $(\eta_i^m)^*$ can be found by solving a set of non-linear fixed-point equations of the form

$$(\eta_i^m)^*=\sum_{\substack{\ell=1\\ \ell\neq i}}^{S}\sum_{k\in\mathcal{F}_\ell\cap\mathcal{V}_m}\underset{r^\ell}{E}\{\hat{\gamma}^*_{k\ell}(r^\ell,\{(\eta_j^n)^*, n=1,\dots,M, j=1,\dots,S; j\neq\ell\})\},$$

$$m=1,\dots,M, i=1,\dots,S \tag{17}$$

The procedure for finding p.b.p.o. strategies for the problem outlined by (10)-(14) can then be split into two parts: i) first, having fixed a set of parameters $(\eta_i^m)^*, m=1,\dots,M, i=1,\dots,S$, we derive the analytical expression of strategies (15); ii) subsequently, we seek a numerical solution to the fixed-point equations (17). Since the mathematical problem outlined here is equivalent to the one that we solved in [21], we do not repeat the derivation; rather, we provide the analytical form and the algorithmic description of the p.b.p.o. solutions, which will be used to derive the numerical results in Section VI. By defining a suitable matrix A and a vector $b_i^*=\left[b_i^{1*},\dots,b_i^{F_i*}\right]^T$, and dropping both index i and superscript *, we can write DMi's optimization problem (16) in the more compact form

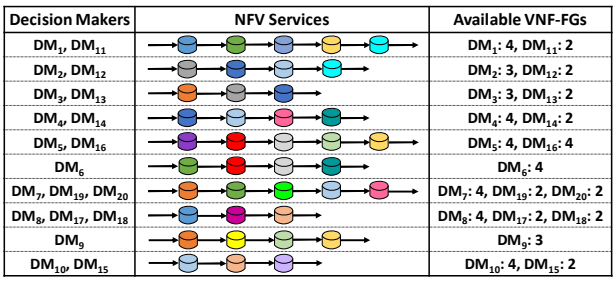
$$\min_u\left[\frac{1}{2}u^TAu+b^Tu\right] \tag{18}$$

subject to

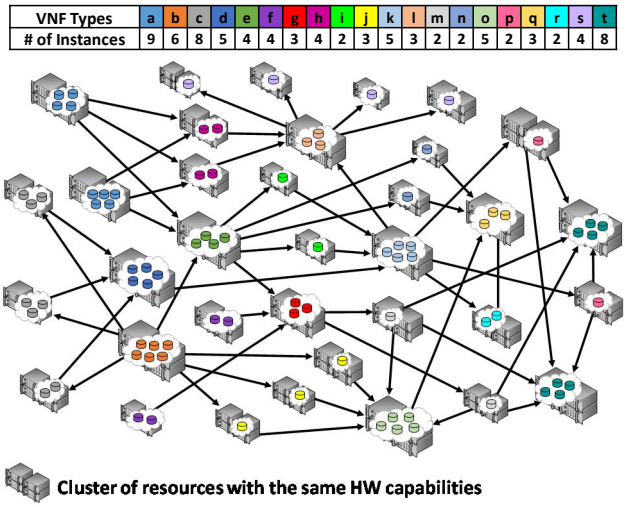
$$u^T\cdot\underline{1}=r\,;\ \ u\geq 0 \tag{19}$$

where $\underline{1}$ is a column vector of all 1s. A is a $F\times F$ symmetric matrix. In order to ensure that matrix A in (19) is positive definite, we assume each VNF used by a PoP to have at least a VM unshared with other VNFs used by the same PoP. Let $[0,R]$ be the support of the probability distribution of r.

It is shown in [21] that the solution for any value of $r\in[0,R]$ is piecewise linear in a number of sub-intervals $[r_x,r_x+1]\subseteq[0,R]$, constituting a partition of $[0,R]$, with non-zero linear terms for $r\in[r_x,r_x+1]$ taking on the form



(a) NFV service specifications



(b) NFV network

Fig. 2. Scenario with 20 DMs, each one accessing an NFV service through a given number of available VNF-FGs over the shared NFV network.

$$u_x^*(r)=\frac{\tilde{A}_x^{-1}\underline{1}_x}{\underline{1}_x{}^T\tilde{A}_x^{-1}\underline{1}_x}(r-r_x) \tag{20}$$

where $\underline{1}_x$ and $\tilde{A}_x$ are the unity sub–vector and the sub–matrix of A that take into account only the indexes in the subset $I'_x\subseteq I_x$, such that: i) the indexes in I_x correspond to the minimal and equal components of vector $b_x=\tilde{A}_{x-1}u^*_{x-1}(r_x)+b_{x-1}\geq 0$ (which is the gradient of the cost functional in (18) computed at the optimal solution $u^*_{x-1}(r_x)$), with corresponding matrix A_x, which would give rise to the control vector $u_x(r)=\frac{A_x^{-1}\underline{1}}{\underline{1}^TA_x^{-1}\underline{1}}(r-r_x)$ ii) the corresponding components of the control vector $u_x(r)$ are non-negative. The solutions in the various intervals can be calculated recursively, given that, for the first inerval $r_0=0$, $u_0^*(0)=0$, $A_0=A$, and $b_0=b$. The range of validity of the solution within each interval is determined for increasing r either by the first point (if any) where a positive but decreasing control variable becomes zero, or by the first point (if any) where a gradient component originally non–minimal becomes part of the minimal ones.

We finally note that the team solutions we can find by applying the outlined methodology would coincide with a unique team-optimal solution only in case of existence of a unique fixed point for equations (17), whose investigation, however, is beyond the scope of this work. Optimality of the strategies we have derived is therefore assured only in p.b.p.o. sense.

V. Performance Evaluation

To evaluate the performance of the proposed load balancing approach, we consider the scenario in Fig. 2, and find the p.b.p. optimal strategies of the DMs by using the numerical method in [21]. Then, we compare the normalized dynamic power consumption induced by the VNF-FGs with the team and the uniform flow distribution, as well as with the one corresponding to concentrating each DM's load on their respective least-cost paths.

In more detail, we evaluate a system with 20 DMs, each one accessing a specific NFV service over a shared network of 35 resource clusters, hosting 84 VNF instances of 20 VNF types, as illustrated in Fig. 2. Generally, instances of the same VNF type can be hosted among resources with the same or varying capabilities – this is captured in the considered scenario through clustering of resources and various combinations of HW parameters. Moreover, some DMs accessing the same NFV service can have different number of available VNF-FGs (i.e., paths), simulating a more general scenario with differentiated services.

The (physical/logical) links served by the VMs $\{j\}$ (i.e., VNF implementations) in the system are identified with an index j, and a weight coefficient w_j that depends on a number of components. Particularly, w_j is given by the sum of the coefficient h_j of the serving VM, and of a random number generated from the uniform distribution, $\mathcal{U}(0,10)$. The former is determined by the underlying HW (i.e., K_j and α_j parameters), and the degree of parallelizability of the VNF code (i.e., c_j and $\{\beta_n^{(j)}\}$ information), while the latter accounts for the different network paths traversed by the contributing flows that enter the VM, as noted in Section IV; here we decided for a random choice, to avoid being bound to a fixed network topology.

Following [12], in which $K_j \in \mathcal{U}_K = \mathcal{U}(1,10)$ and $\alpha_j \in \mathcal{U}_\alpha = \mathcal{U}(2,3)$, we consider two cases in this work: (a) *homogeneous* HW, where $K_j = K \in \mathcal{U}_K$, $\alpha_j = \alpha \in \mathcal{U}_\alpha$, $\forall j$, and; (b) *heterogeneous* HW, where the parameters among VMs $\{j\}$ hosted on the resource cluster σ are generated as $K_j = K_\sigma \in \mathcal{U}_K$, $\alpha_j = \alpha_\sigma \in \mathcal{U}_\alpha$, $\forall j \mapsto \sigma$, $\sigma = 1, \dots, 35$. The latter is supposed to cover not only the possible HW heterogeneity inside a PoP (e.g., server level), but also the scenario where the VNF-FG spans multiple PoPs of varying capabilities.

The parallelizability of each VNF instance in the system (even the ones performing the same functionality) is generated randomly, supposing that a VNF code is parallelizable into 1, 2, 4, 6 or 8 vCPUs. Random permutation is used to generate varying $\{\beta_n^{(j)}\}$ values, even with the same c_j; this emulates the performance variations of a code on different execution environments. For the sake of simplicity, but without loss of generality, we suppose that the 20 DMs have the same maximum load R_{max}, and their instantaneous loads $r(\mathbf{DM}_i)$, $i = 1, \dots, 20$, are uniformly distributed in $[0, R_{max}]$.

A. Team-optimal Load Balancing

In both homogeneous and heterogeneous HW cases, the resulting p.b.p.o. load distribution policies of the DMs highly depend on w_j and the number of VNF-FGs sharing a VM. Some DMs have relatively "static" strategies, allocating a constant fraction of their load to the (or a subset of) available VNF-FGs, while others result in more interesting strategies, adapting their distributions with the load. Fig. 3 shows examples of p.b.p.o. load distribution policies of **DM4** and **DM14**, where paths indicated with the same color correspond to the same VNF-FGs (e.g., Path3(**DM4**) and Path2(**DM14**)). It can be observed that the general form of the team-optimal solutions is piecewise-linear.

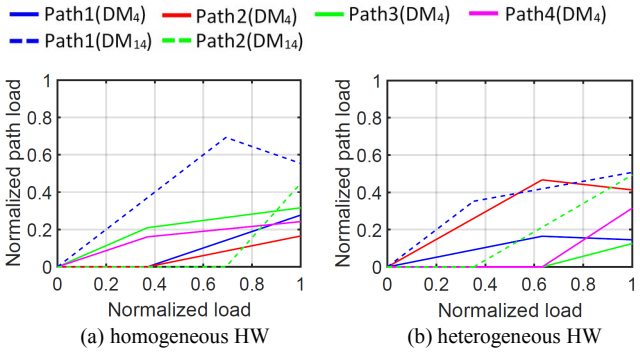


Fig. 3. P.b.p. optimal load distribution policy of **DM4** and **DM14**.

B. Dynamic Power Consumption

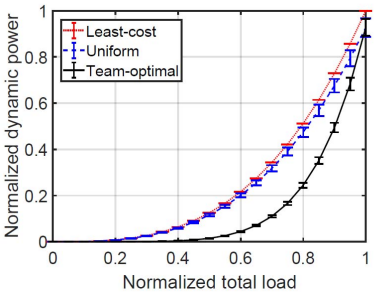
Here, we evaluate the normalized dynamic power consumption induced by the 20 DMs when the team solutions are applied in the homogeneous and heterogeneous HW cases. As comparison, two baseline policies are considered in this work: (a) *least-cost* path, in which the DMs route all the load to the VNF-FG with the minimum execution cost (i.e., as a sum of the link weights) among the available paths, and; (b) *uniform* flow distribution, in which all available paths are allocated equal fractions of the load. To add statistical significance in the results, 95% confidence intervals are obtained from 10 runs of varying seeds.

As shown in Fig. 4, the p.b.p.o. team solutions gave better performance in terms of energy saving with respect to the two baselines, achieving improvements of up to over 45% to around 4 orders of magnitude when the normalized total load is less than 80%. While similar behaviors can be observed with both homogeneous and heterogeneous HW, with the latter there are some cases in which uniform load distribution result in the highest consumption, rather than the one using only the least-cost paths (which is always the case for the former). This can be expected especially when the costs of a DM's available VNF-FGs vary greatly.

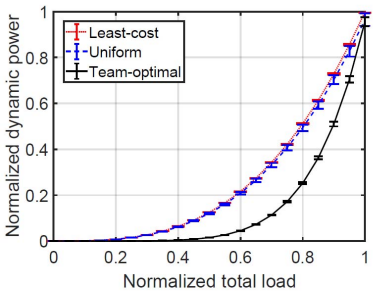
Though in a relatively simple topology of VNF-FGs, the results highlight some of characteristics that can be expected by the application of p.b.p.o. team strategies in the NFV environment we have considered. In particular, in the presence of different types of heterogeneous HW and multiple interactions among DMs' paths, we expect higher energy saving gains.

VI. Conclusions

We have formulated a quadratic constrained team optimization problem in a network virtualization environment characterized by the presence of multiple VNF-FGs, offered



(a) homogeneous HW



(b) heterogeneous HW

Fig. 4. Normalized dynamic power consumption induced by the VNF-FGs when the least-cost, uniform and p.b.p. optimal load distribution policies are applied.

to their users (substantially, slice instances serving specific traffic aggregation points, like PoP datacenters) on VMs realized above a general-purpose HW infrastructure. The quadratic form of the team cost stems from taking into account the effect of energy-saving policies applied by the Infrastructure Providers which are the owners of the HW, where energy is effectively consumed. In this respect, it is worth noting that, in the absence of the interaction effected by the cost function, IPr's tenants (the Network Service Providers), which operate in a completely virtualized environment, would have neither a direct perception (e.g., based on measurements) of the energy aspect, nor any incentive to be aware of it.

The solution to the team optimization problem has been provided analytically in the form of parametrized p.b.p.o. strategies that turn out to be piecewise linear in the workload of each specific DM. As such aggregated workloads can vary dynamically over relatively short time scales (e.g., in the order of a few seconds, depending on end users' density and mobility), informationally decentralized strategies lend themselves to fast reaction without the need of additional signaling. The form of the p.b.p.o. team strategies has been found numerically in a simple example, and their effect on the energy consumption has been investigated and compared with least-cost and uniform distributions of the load.

ACKNOWLEDGMENT

This work was supported by the European Commission in the framework of the H2020 5G-PPP MATILDA Project (contract no. 761898).

REFERENCES

[1] A. Manzalini *et al.*, "Towards 5G Software-Defined Ecosystems – Technical Challenges, Business Sustainability and Policy Issues," IEEE SDN Initiative Whitepaper, July 2016; http://resourcecenter.fd.ieee.org/fd/product/white-papers/FDSDNWP0002.

[2] 5GPPP Architecture Working Group, View on 5G Architecture, Version 2.0, Dec. 2017. Online: https://5g-ppp.eu/wp-content/uploads/2018/01/5G-PPP-5G-Architecture-White-Paper-Jan-2018-v2.0.pdf.

[3] S. Thalanany, P. Hedman, "Description of Network Slicing Concept," NGMN 5G P1 Requirements & Architecture, Work Stream End-to-End Architecture, version 1.0.8, Sept. 2016. Online: https://www.ngmn.org/uploads/media/161010_NGMN_Network_Slicing_framework_v1.0.8.pdf [last access 23rd March 2018].

[4] 3GPP, "Study on Management and Orchestration of Network Slicing for Next Generation Network," TR 28.801, version 15.0.0, Sept. 2017.

[5] J. Ordóñez Lucena, P. Ameigeiras, D. Lopez, J. J. Ramos-Muñoz, J. Lorca, J. Folgueira, "Network Slicing for 5G with SDN/NFV: Concepts, Architectures, and Challenges," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 80-87, May 2017.

[6] E. Hernandez-Valencia, S. Izzo, B. Polonsky, "How Will NFV/SDN Transform Service Provider OpEx?," *IEEE Netw.*, vol. 29, no. 3, pp. 60-67, May/June 2015.

[7] D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, S. Uhlig, "Software-Defined Networking: A Comprehensive Survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14-76, Jan. 2015.

[8] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. De Turck, R. Boutaba "Network Function Virtualization: State-of-the-art and Research Challenges," *IEEE Commun. Surv. & Tut.*, vol. 18, no. 1, pp. 236-262, 1st Qr. 2016.

[9] R. Bolla, R. Bruschi, F. Davoli, C. Lombardo, J. F. Pajo, O. R. Sanchez, "The Dark Side of Network Functions Virtualization: A Perspective on the Technological Sustainability," *Proc. IEEE Int. Conf. Commun. (ICC 2017)*, Paris, France, May 2017.

[10] R. Bolla, R. Bruschi, F. Davoli, F. Cucchietti, "Energy Efficiency in the Future Internet: A Survey of Existing Approaches and Trends in Energy-Aware Fixed Network Infrastructures," *IEEE Commun. Surveys Tuts.*, vol. 13, no. 2, pp. 223-244, 2nd Qr. 2011.

[11] Network Functions Virtualisation (NFV); Architectural Framework, ETSI GS NFV 002 V1.1.1 (2013-10).

[12] R. Bruschi, A. Carrega, F. Davoli, "A Game for Energy-Aware Allocation of Virtualized Network Functions," *J. Electr. Comput. Eng.*, vol. 2016, Article ID 4067186, Feb. 2016; http://dx.doi.org/10.1155/2016/4067186.

[13] Network Functions Virtualisation (NFV); Management and Orchestration, ETSI GS NFV-MAN 001 V1.1.1 (2014-12).

[14] P. Mell, T. Grance, "The NIST definition of Cloud Computing," National Institute of Standards and Technology, U.S. Department of Commerce, Special Publication 800-145, Sept. 2011, Online: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf [last access 23rd March 2018].

[15] L. M. Vaquero, L. Rodero-Merino, and R. Buyya, "Dynamically Scaling Applications in the Cloud," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 1, pp. 45-52, Jan. 2011.

[16] A. A. Mann, "Allocation of Virtual Machines in Cloud Data Centers—A Survey of Problem Models and Optimization Algorithms," *ACM Comput. Surveys*, vol. 48, no. 1, Sept. 2015.

[17] S. Yüksel, T. Başar, Stochastic Networked Control Systems—Stabilization and Optimization under Information Constraints, Birkhäuser, New York, NY, USA, 2013.

[18] J. Marshak, R. Radner, *The Economic Theory of Teams*, Yale University Press, New Haven, CT, 1971.

[19] R. Bolla, R. Bruschi, F. Davoli, P. Lago, "Optimizing Power Delay Product in Energy-Aware Packet Forwarding Engines," *Proc. 24th Tyrrhenian Int. Workshop Digit. Commun. (TIWDC 2013) – Green ICT*, Genoa, Italy, Sept. 2013.

[20] A. S. Cassidy, A. G. Andreou, "Beyond Amdahl's Law: An Objective Function that Links Multiprocessor Performance Gains to Delay and Energy," *IEEE Trans. Comput.*, vol. 61, no. 8, pp. 1110–1126, Aug. 2012.

[21] M. Aicardi, R. Bruschi, F. Davoli, P. Lago, "A Decentralized Team Routing Strategy Among Telecom Operators in an Energy-Aware Network," *Proc. SIAM Conf. Control Appl.*, Paris, France, July 2015, pp. 340-347.