Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 110 (2017) 352–360

Procedia Computer Science

www.elsevier.com/locate/procedia

The 12th International Conference on Future Networks and Communications (FNC-2017)

Orchestrating Softwarized Networks with a Marketplace Approach

Salvatore D'Oro, Sergio Palazzo, Giovanni Schembra*

University of Catania, Catania, Italy

Abstract

In the last years, network softwarization is gaining increasing popularity since it allows to achieve dinamicity and flexibility in network management, stimulating a lot of interest by both academia and industry. Cloud computing paradigm together with the new networking paradigms of Software Defined Networking (SDN) and Network Function Virtualization (NFV) are supporting this evolution, by providing network services as single Virtual Network Functions (VNFs) or chains of them. The main problem is scalability of both infrastructure and management. In fact, in order to support the SDN/NFV paradigm, the Telco Operator should deploy huge data centers, which have to be geographically distributed to guarantee low latencies to time-constrained flows, and implement complex orchestration policies. To this purpose, this paper proposes to extend the SDN/NFV framework with a marketplace where Telco Operator customers behave as third-party sellers with their hardware and software resources providing VNF as a service (VNFaaS), so helping the Telco Operator in providing network services in an efficient and scalable way.

Peer-review under responsibility of the Conference Program Chairs.

Keywords: Softwarized Networks, Orchestration, Resource Management, Marketplace Model, Game Theory.

1. Introduction

In the last decade, the Internet has registered a tremendous increasing in capillarity and networked devices such that the current network infrastructure can hardly meet the demands of network development. In this evolution, network softwarization, Software Defined Networking (SDN) and Network Functions Virtualization (NFV) [1] have become the focus of networking research and development worldwide.

Telco Operators (TO) have shown a great interest in supporting the evolution of this network softwarization process, thanks to its advantages in realizing more flexible networks where services can be instantly monitored, controlled, billed, and managed on the fly. Key elements for the design of these systems are allocation, management and orchestration of network resources, that result more challenging as compared to scenarios of legacy networks.

In this context, the main problem stays in the difficulties in deciding how many instances using for each virtual network function (VNF), how many servers, in the following referred to as *VNF Servers*, maintaining active in the network, which servers using to host the instances of each VNF, and the amount of hardware resources dedicating in

* Corresponding author. Tel.: +39-095-7167389.
E-mail address: schembra@dieei.unict.it

1877-0509
Peer-review under responsibility of the Conference Program Chairs.
10.1016/j.procs.2017.06.077

each server in terms of computing, storage and networking. An additional problem, referred to as the *Service Chain Composition problem*[2], is how chaining VNF instances to realize more complex network services (NSs). This choice has to be done for each flow entering the network, and taking into account the users' requirements in terms of both required quality of service (QoS) and payment availability.

It is well-known that the service chain composition problem is NP-hard[3], and many optimal and sub-optimal solutions have been proposed in the previous literature[4,5,6,7]. However, those approaches are not well-suited to dynamic scenarios where the status of the network varies in time. Furthermore, those solutions often requires perfect knowledge, a condition which might not always be satisfied, and do not consider the competitive interactions among the service providers which are selfish and profit-maximizing.

To this purpose, starting from our previous work[8], this paper proposes to introduce the concept of *marketplace* into the NFV market, where VNF provision and network orchestration are centralized by the TO (see Fig. 1). According to the marketplace definition[9], customers of the TO can participate as third-party sellers with their hardware and software resources by offering VNFs as a service (VNFaaS), so helping the TO in providing NSs in an efficient and scalable way. In addition, in order to achieve scalability also in terms of management, unlike the classical approaches that entirely concentrate the complex tasks of orchestration and resource allocation into a single entity, i.e. the *Orchestrator*, here we propose, in line with the definition of marketplace, a distributed solution to the problem, where each user finds by itself, for each of its flows, the best service chain of VNF instances that accommodate its individual requirements. In particular, we consider NFV-specific network requirements, such as the congestion level on VNF servers, the latency incurred by traffic flows, and the price charged by VNF Servers to execute VNFs.

In order to allow users to autonomously compose the VNF chain that better satisfies their requirements, we will introduce the *Network Service Broker* (NSB), an entity aimed at representing one flow or an aggregate of flows with the same QoS and price requirements, and generated by users belonging to the same portion of network, in the following referred to as *NSB Scope*. Therefore, the marketplace actors are: 1) the VNF Servers, that are the sellers of the VNFs; 2) the Users, whose flows are represented by the NSBs, and that play the role of buyers; 3) the TO that, through the Network Orchestrator, coordinates the whole system.

Servers autonomously decide the price to be applied to each VNFs. NSBs, on the other hand, in order to compose a structured NS for the flows they are in charge to manage, choose one VNF instance for each VNF of the service chain realizing the required NS, and this is done according to the price specified by each Server and the corresponding expected performance in terms of both experienced latency and provided resources. In this way the task of associating each flow to a service chain is not decided by the Orchestrator, but is obtained in an autonomous and distributed way, as a consequence of the interaction between Users and VNF Servers. This interaction is performed by way of the Orchestrator not only to exchange the needed information, but preserving privacy for all the market place participants. Interactions between Servers and Users are modeled by leveraging on the game theory. In[8], we focused on a transitory analysis of the marketplace. In this paper, instead, besides a more detailed description of the system architecture, we extend the results in[8] by presenting an efficiency analysis of the proposed solution, and a steady-state analysis of the main parameters of the marketplace model such as prices, the number of users that decide to use each VNF Server, and the achieved profit.

The rest of the paper is organized as follows. In Section 2, we describe the reference system. In Section 3, the Marketplace model is discussed. In Section 4, the numerical results are shown. Conclusions are drawn in Section 5.

2. Reference System

In this section we describe the reference scenario, constituted by an SDN/NFV network of a TO. A NS is realized by one or more VNFs organized in chain, according to VNF Forwarding Graph (VNF FG)[10,11]. The main system players are the Users, the VNF Servers and the Orchestrator.

According to the flow type and its importance, Users request a specific NS for each of their flows. For example, different network services can be requested for video streaming and email traffic flows. Moreover, even two video streaming flows may require different levels of QoS, according to their importance and the willingness of the customers to pay no more than a given price.

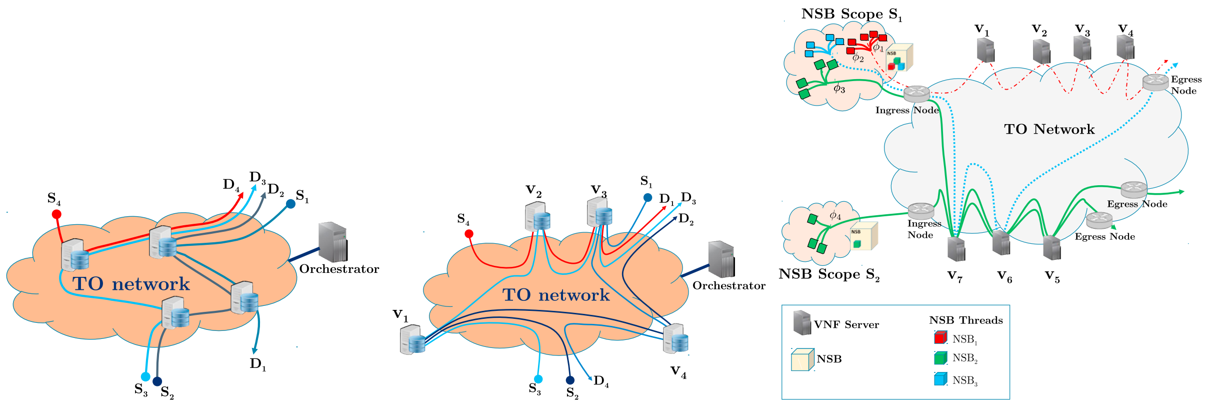


Fig. 1. a) Centralized market scenario, where VNFs are provided by the TO; b) Proposed marketplace scenario, where VNFs are provided by third-party servers; c) An illustrative example of NSBs and their scopes.

The role of *VNF Servers* is played by customers of the TO network that enter the NFV marketplace with some VNF, in order to obtain some economic benefits by serving the TO network. The price of each VNF instance provided by a VNF Server is decided autonomously.

The *Network Orchestrator* is implemented according to the Management and Orchestration (MANO) specifications [12], and is responsible of management and orchestration of the whole system. For NSs constituted by only one VNF, the Network Orchestrator decides the number of VNF instances to be run in the network, and their placement. Likewise, in the case of NSs realized as chains of more than one VNF, the Network Orchestrator is also in charge of chaining the running VNF instances to realize instances of networks services with different levels of QoS, i.e. the so-called Service Chain Composition task. This is a hard multi-objective optimization problem that should be solved by the Network Orchestrator at run-time. With this in mind, this paper approaches this problem by introducing the marketplace model, in such a way that this task is no longer centralized in the single entity of the Network Orchestrator, but distributed to the network customers. In addition, in this paper we introduce the role of NS broker, with the purpose of relieving Users to directly compose by themselves the VNF chains. This role is played by the NSBs, software applications that represent the flows generated by Users accessing the Internet from nodes belonging to a delimited portion of network, in the following referred to as NSB Scope.

In the sequel, we will indicate as a *traffic flow* either a single traffic flow, or an aggregate of them that, although transmitted by different Users, or generated by different applications of the same User, have the same ingress and egress edge nodes, and require the same VNF chain with the quality requirements. Fig. 1(c) shows an example highlighting two NSB Scopes, S_1 and S_2, the first with three flows, ϕ_1, ϕ_2, and ϕ_3, and the second with only one flow, ϕ_4. Therefore, three different NSB threads run in the NSB Scope S_1, while one NSB thread works in the NSB Scope S_2. Since the flow ϕ_4 coming from S_2 requires the same NS of one of the flows from S_1, i.e. ϕ_3, the two threads labeled as NSB_2 in Fig. 1(c) (one in each NSB Scope) compete with each other as players of the same game.

The game modeling the interaction among NSBs will be described in the following section, as well as the interaction among VNF Servers.

3. The VNF Marketplace

Before describing the model of the VNF marketplace, let us introduce some notation. Let $\mathcal{F}^{(all)}$ be the set of VNFs that are available in the marketplace. As described so far, the entities that participate in the marketplace for the generic VNF $f \in \mathcal{F}^{(all)}$ are the VNF Servers that run f, and the NSBs that need f to be included in the chains for the flows they manage. The former are the sellers, while the latter behave as buyers. The other entity that plays an important role in the marketplace is the TO that, besides providing interconnection service to its customers, has decided to include the VNF f in its catalogue in order to gain some economic benefit from it.

Decisions taken by VNF Servers and NSBs depend on both individualistic interests, e.g. maximize (minimize) their own utility (cost), and decisions taken by counterparts, e.g. opponents' strategies. For example, NSBs decide to include a VNF instance provided by one specific VNF Server depending on the proposed price and expected communication delay. On the contrary, VNF Servers aiming at maximizing their revenues, are not likely to cooperate with each other, and their actions depend on the number of flows that are using VNFs provided by them.

In order to simplify the notation, in the following we will focus on one NS, indicated as $\mathcal{Y}$. We will also use the same symbol $\mathcal{Y}$ to indicate the ordered sequence of the VNFs constituting it, i.e. $\mathcal{Y} = \{1, 2, \dots, F\}$, F being the number of VNFs in the chain.

We assume that VNFs are executed on a set $\mathcal{S}$ of V VNF Servers. For each VNF Server $v \in \mathcal{S}$, let $\mathcal{Y}_v \subseteq \mathcal{Y}$ be the subset of VNFs provided by v. The set of NSBs interested to the NS $\mathcal{Y}$ will be referred to as $\mathcal{U}$, let N be its cardinality. Different NSBs handle different flows, and are located at different network areas. As said in the previous section, we assume that each NSB manages only one flow that can be either a single flow, or an aggregate of flows. Thus, let λ_i be the bit rate of the flow handled by the NSB $i \in \mathcal{U}$, and let s_i and t_i be the ingress and the egress nodes of that node. Hence, we associate to each NSB a pair (s_i, t_i), representing the network path of the flow managed by it.

In order to describe the chains that each NSB can compose, let $\mathcal{W}$ be the set of all possible configurations for the NS $\mathcal{Y}$. The generic element $\mathbf{w}_i \in \mathcal{W}$ is a specific NS configuration chosen by the NSB representing the flow $i \in \mathcal{U}$. More in deep, $\mathbf{w}_i$ is defined as the F-tuple $\mathbf{w}_i = (w_i(1), w_i(2), \dots, w_i(F))$, where $w_i(f) \in \mathcal{S}$ is the VNF Server which has been chosen by the NSB i to execute the function $f \in \mathcal{Y}$. Moreover, since different flows can use a same instance of the VNF f, they will experience a congestion level due to the load on the VNF Server where f is running. Therefore, in order to represent the congestion level for each VNF when we focus on a specific NSB i, we need to consider the choices of the other NSBs. To this end, let $\mathbf{w}_{-i}$ be the set of all the NS configurations chosen by the NSBs in $\mathcal{U} \backslash \{i\}$, that is, $\mathbf{w}_{-i} = (\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_{i-1}, \mathbf{w}_{i+1}, \dots, \mathbf{w}_N)$.

Now, we can define the total cost experienced by the flow represented by the NSB $i \in \mathcal{U}$ when it decides a given configuration $\mathbf{w}_i$, while the other NSBs use the configurations represented by $\mathbf{w}_{-i}$. Taking into account that it depends on the congestion level of the VNF Servers running the chosen VNFs, the price applied by them, and the delay encountered along the chosen path, we define the following *cost function*:

$$C_i(\mathbf{w}_i, \mathbf{w}_{-i}) = c^{(C)}(\mathbf{w}_i, \mathbf{w}_{-i}) + \gamma_i \left(c_i^{(\mathcal{T})}(\mathbf{w}_i) + \beta_i c_i^{(\mathcal{P})}(\mathbf{w}_i) \right) \tag{1}$$

where γ_i and β_i are two non-negative parameters decided by the NSB i to weigh the above three contributions.

In order to characterize the first contribution, let us define the set of NSBs in $\mathcal{U}$ which have chosen the server $v \in \mathcal{S}$ to receive the function $f \in \mathcal{Y}$:

$$\Gamma^f(\mathbf{w}_i, \mathbf{w}_{-i}, v) = \left\{ j \in \mathcal{U} : w_j(f) = v, w_j(f) \in \mathbf{w}_j, \mathbf{w}_j \in (\mathbf{w}_i, \mathbf{w}_{-i}) \right\} \tag{2}$$

where $\mathbf{w}_j$ indicates the NS configuration chosen by the NSB j, while $(\mathbf{w}_i, \mathbf{w}_{-i}) \in \mathcal{W}^N$ represents the set of the NS configurations chosen for all the NSBs requiring the considered network service $\mathcal{Y}$. The term $\mathcal{W}^N$ stands for the N-ary Cartesian power of the set $\mathcal{W}$. Accordingly, we have that the congestion level experienced by the flow corresponding to the NSB i for function f on the chosen server $w_i(f)$ is

$$\delta_f(\mathbf{w}_i, \mathbf{w}_{-i}) = \sum_{j \in \Gamma^f(\mathbf{w}_i, \mathbf{w}_{-i}, w_i(f))} \lambda_j \tag{3}$$

where $\delta_f(\mathbf{w}_i, \mathbf{w}_{-i}) = \delta_f(\mathbf{w}_j, \mathbf{w}_{-j})$ if $w_j(f) = w_i(f)$. Finally, thanks to (3), it follows that the overall congestion level experienced by the flow of NSB $i \in \mathcal{U}$ on the whole service chain when the other NSBs have chosen the NS configurations described by $\mathbf{w}_{-i}$, is

$$c_i^{(C)}(\mathbf{w}_i, \mathbf{w}_{-i}) = \sum_{f=1}^{F} \delta_f(\mathbf{w}_i, \mathbf{w}_{-i}) \tag{4}$$

Now, in order to characterize the end-to-end delay contribution to the cost function defined in (1), we describe the communication latencies between NSBs and VNF Servers with the following three latency matrices. Let $\mathbf{D} = (d_{v',v''})_{v',v'' \in \mathcal{S}}$ be the *Server-to-Server latency matrix* containing the latencies between all VNF Servers involved in the NS $\mathcal{Y}$. Similarly, let $\mathbf{D}^{\text{in}} = \left(d^{\text{in}}_{s_i,v} \right)_{i \in \mathcal{U}, v \in \mathcal{S}}$ be the *ingress-to-Server latency matrix* which contains the latencies between

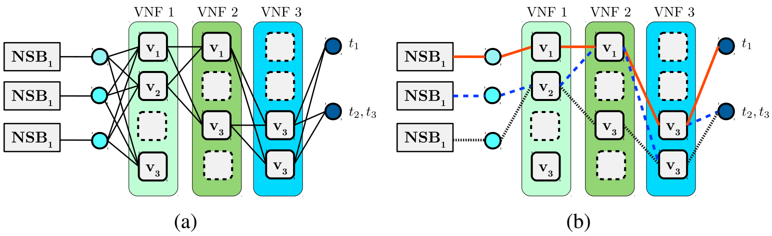


Fig. 2. a) Service Chain Model with $F = 3$, $V = 4$ and $N = 3$; b) An illustrative example of a possible service chain configuration.

all ingress nodes and VNF Servers. Finally, let $\mathbf{D}^{\text{out}} = \left(d^{\text{out}}_{v,t_i}\right)_{v\in\mathcal{S},i\in\mathcal{U}}$ be the *Server-to-egress latency matrix* containing all latencies between VNF Servers and egress nodes. Therefore, for a given NS configuration $\mathbf{w}_i$ chosen by NSB i, the end-to-end latency contribution in (1) can be defined as follows:

$$c_i^{(\mathcal{T})}(\mathbf{w}_i) = d_{i,w_i(1)} + \sum_{f=2}^{F} d_{w_i(f-1),w_i(f)} + d_{w_i(F),i} \tag{5}$$

where the term $d_{s_i,w_i(1)} \in \mathbf{D}^{\text{in}}$ is the latency between the ingress node s_i and the server $w_i(1) \in \mathcal{S}$, the term $d_{w_i(f-1),w_i(f)} \in \mathbf{D}$ is the inter-server latency between servers $w_i(f-1)$ and $w_i(f)$, and the term $d_{w_i(F),t_i} \in \mathbf{D}^{\text{out}}$ is the latency between the server $w_i(F)$ and the egress node t_i.

Finally, let us describe the price contribution. To this purpose, as usual in a marketplace approach, we assume that the *VNF price* for an instance of a VNF $f \in \mathcal{Y}_v$ provided by a VNF server v is decided by that server. We will indicate this price as $p_{v,f}$. Therefore, the price contribution $c_i^{(\mathcal{P})}(\mathbf{w}_i)$ to a given NS configuration $\mathbf{w}_i$, is given by

$$c_i^{(\mathcal{P})}(\mathbf{w}_i) = \sum_{f=1}^{F} p_{w_i(f),f} \tag{6}$$

An example of the considered service chain model where $F = 3$ VNFs compose the NS $\mathcal{Y} = \{1, 2, 3\}$, $V = 4$ VNF Servers are in the network, and $N = 3$ NSBs choose the NS configuration for their flows is presented in Fig. 2(a). As shown in Fig. 2(a), some servers, highlighted with dashed border lines, do not provide functions in the chain. Also, a subset of NSBs in $\mathcal{U}$ could possibly share the same ingress and/or egress nodes, i.e., NSBs $i = 2, 3$ that have the same egress node $t_2 = t_3$, even if they have different ingress nodes $s_2 \neq s_3$. Moreover, in Fig. 2(b), we show an illustrative example where we explain how service chaining is performed in the considered scenario. The service chains chosen for the three flows, $\mathbf{w}_1$, $\mathbf{w}_2$ and $\mathbf{w}_3$, are represented by solid, dotted and dashed lines, respectively. As shown in Fig. 2(b), flows 2 and 3 share the same servers v_2 and v_4 to execute the VNFs 1 and 3, respectively. Accordingly, the server load at v_2 to execute VNF 1 and at v_4 to execute VNF 3 is equal to $\lambda_2 + \lambda_3$. Similarly, the server load on v_1 to execute VNF 1 is equal to λ_1. The same also holds for v_3 which is selected by the flow 1 to execute VNF 3. Finally, the server load on v_1 with respect to VNF 2 is equal to $\lambda_1 + \lambda_2$.

In the next subsections we will describe the interactions among NSBs and the ones among VNF Servers. Specifically, Section 3.1 we will describe the Service Chain Composition problem as a congestion game[13], while a simple, but effective, pricing mechanism is presented in Section 3.2 to model the interactions among the VNF Servers.

3.1. NSB interaction model

It has been shown that the selfish and conflictual interactions between NSBs can be modeled as the following weighted congestion game $\mathcal{G}$[14]:

$$\mathcal{G} = \left(\mathcal{U}, (\lambda_i)_{i\in\mathcal{U}}, \mathcal{S}_f, \mathcal{W}^N, (C_i)_{i\in\mathcal{U}}\right) \tag{7}$$

where $\mathcal{U}$ is the set of *players*, $\mathcal{S}_f$ is the set of servers which provide VNF $f \in \mathcal{Y}$, and represents the set of *resources* of the game, $\mathcal{W}$ are all the possible NS configurations, C_i is the cost function of the i-th NSB, and the bit rates λ_i of the relative flows are the *weights* of the congestion game. In the sequel, we will refer to the VNF Servers as the resources of the congestion games, and we use terms NSB and player interchangeably.

One major question about game $\mathcal{G}$ is if it possesses a Nash Equilibrium (NE), defined as a strategy profile $(\mathbf{w}_1^*, \mathbf{w}_2^*, \ldots, \mathbf{w}_N^*) \in \mathcal{W}^N$ where no player has incentive to deviate unilaterally. Also, if a NE exists, it is of extreme importance to provide an effective method to compute it, and to investigate its efficiency with respect to centralized optimal solutions. It is easy to demonstrate[14] the existence of a NE but, more importantly, the boundedness of the Price of Anarchy (PoA), which is used to measure the efficiency of the NS configuration at the NE with respect to the optimal NS configuration. Furthermore, by exploiting the fully-distributed and privacy-preserving unilateral service chain selection algorithm proposed in our previous work[14], we achieve a highly-scalable solution for the service chain composition problem. Specifically, the considered algorithm converges with polynomial computational complexity $O(N^2F^2V^3)$ towards a NE.

3.2. VNF Server pricing model

To handle and process flows traversing a given VNF Server, the allocation of a proper amount of both computational and storage resources is required. Such an operation generates an incremental cost which, for a given VNF Server $v \in \mathcal{S}$ which handles a flow requiring function f, is here denoted as $\rho_{v,f}$. As an instance, such a cost can be used to express the incremental computation and/or energy cost[15]. It is straightforward to note the the overall handling cost for a VNF Server v to support all the traffic flows is proportional to the number of NSBs $n_{v,f}$ attached to v, that is, $C_{v,f} = \rho_{v,f} \cdot n_{v,f}$. Similarly, the revenue for the VNF Server v to provide a VNF f is proportional to both the number $n_{v,f}$, and the price $p_{v,f}$ applied by this VNF Server, that is, $R_{v,f} = p_{v,f} \cdot n_{v,f}$. Accordingly, we define the *profit function* for a given VNF Server v as follows:

$$\Pi_v^{(\mathcal{S})}(\mathbf{p}_v) = \sum_{f \in \mathcal{Y}_v} \Pi_{v,f}^{(\mathcal{S})}(\mathbf{p}_v) \tag{8}$$

where $\mathbf{p}_v = (p_{v,f})_{f \in \mathcal{Y}_v}$ is the *price vector* for VNF Server v, and $\Pi_{v,f}^{(\mathcal{S})}(\mathbf{p}_v)$ is the profit achieved by the VNF Server v with respect to VNF f, which can be written as

$$\Pi_{v,f}^{(\mathcal{S})}(\mathbf{p}_v) = R_{v,f} - \alpha C_{v,f} = n_{v,f}\left(p_{v,f} - \alpha \rho_{v,f}\right) \tag{9}$$

with α being a non-negative weight which trade-offs between revenues and costs.

We assume that VNF Servers aim at achieving positive profit, thus it follows that the VNF price $p_{v,f}$ must be chosen such that the relationship $\Pi_{v,f}^{(\mathcal{S})}(\mathbf{p}_v) > 0$ holds. Thus, the VNF price has to be higher than the incurred marginal costs, i.e., $p_{v,f} > \alpha\rho_{v,f}$. Moreover, from (9) we have that the $\Pi_v^{(\mathcal{S})}(\mathbf{p}_v)$ depends on the number $n_{v,f}$ of NSBs, whose value is obtained as the outcome of the interactions among NSBs as already described in Section 3.1. Specifically, let $\mathbf{w}^* = (\mathbf{w}_1^*, \mathbf{w}_2^*, \ldots, \mathbf{w}_N^*)$ be the configuration profile at the NE of the congestion game $\mathcal{G}$ when the price profile $\mathbf{p} = (\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_V)$ is applied by the VNF Servers. The number of NSBs $n_{v,f}$ that have chosen the VNF Server v to execute the VNF f under the configuration $\mathbf{w}^*$ is thus defined as $n_{v,f} = |\Gamma^f(\mathbf{w}^*, v)|$, where $\Gamma^f(\mathbf{w}^*, v)$ is defined in (2).

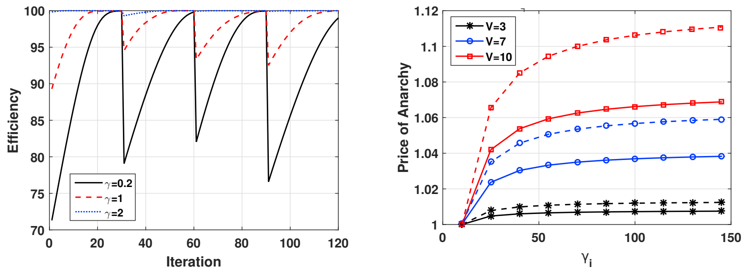
Now, we assume VNF Servers use a stochastic approximation mechanism to update their price $p_{v,f}$ with a periodic update period of Δ seconds. More specifically, for each VNF Server v and VNF f, we assume that the VNF price is updated as follows:

$$p_{v,f}(m+1) = p_{v,f}(m) + \sigma_v(m)\left[n_{v,f}(m) - n_{v,f}(m-1)\right] \tag{10}$$

where m is the generic iteration of the mechanism, and $\sigma_v(m)$ is a decreasing step-size of the stochastic procedure which here is assumed to be defined as $\sigma_v(m) = 1/(m)$ for all $v \in \mathcal{S}$.

(10) shows that if the number of NSBs connected to the VNF Server has been increased in the last iteration, i.e., $n_{v,f}(m) > n_{v,f}(m-1)$, then the price at the following iteration is increased as well, i.e., $p_{v,f}(m+1) > p_{v,f}(m)$. In the opposite case, i.e., if $n_{v,f}(m) < n_{v,f}(m-1)$, the price at the following iteration is decreases, and $p_{v,f}(m+1) < p_{v,f}(m)$.

It is worth noting that, to guarantee a positive profit to each VNF Server, and thus to satisfy the rationality assumption, the relationship $p_{v,f}(m+1) > \alpha\rho_{v,f}$ must be satisfied. However, the stochastic procedure in (10) might violate the above condition by generating a VNF price $p_{v,f}(m+1) < \alpha\rho_{v,f}$. For this reason, we consider a small positive real-valued variable ϵ such that a minimum value $p_{v,f}(m+1) = \alpha\rho_{v,f} + \epsilon$ is considered at each iteration of (10), so ensuring a positive profit to each VNF Server.



(a) (b)

Fig. 3. a) Dynamic evolution of the potential efficiency function ψ; b) PoA as a function of the γ parameter for different values of V and β ($\beta = 5$: solid lines; $\beta = 2$: dashed lines).

4. Numerical Results

In this section, we show the effectiveness of the proposed NFV marketplace model. We have assumed that all flows managed by the NSBs are uniform, that is, for each NSB $i \in \mathcal{U}$, we consider a transmission data rate $\lambda_i = \lambda = 10$ Mbit/s. Furthermore, we assume an NS constituted by a service chain of $F = 5$ VNFs, and we set $\beta_i = \beta = 5$ for all NSBs. The weighing parameter α in (9) is set such that each VNF Server equally weighs revenues and costs, i.e., $\alpha = 1$. Also, we assume that the cost $\rho_{v,f}$ incurred by the VNF Server v to guarantee the required resources to a new flow requiring function f is $\rho_{v,f} = 10$ price units (PU) for all $v \in \mathcal{S}$ and $f \in \mathcal{Y}_v$.

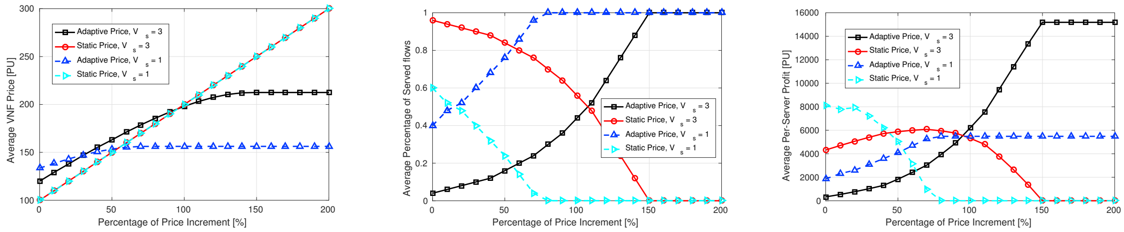
Moreover, we assume that the VNF Servers randomly decide a price $p_{v,f}$ uniformly distributed in the interval $[1, 500]$, and the step-size in (10) is $\sigma_v(m) = 1/(m)$ for all $v \in \mathcal{S}$. Finally, the elements of the latency matrices $\mathbf{D}$, $\mathbf{D}^{\text{in}}$ and $\mathbf{D}^{\text{out}}$ are generated according to a Gamma distribution with mean value of 8 ms and variance equal to 0.004 ms [14]. The results presented are averaged over 100 simulation runs.

Let the *potential efficiency* function ψ be defined as $\psi(\mathbf{w}) = \frac{\Phi(\mathbf{w}^*)}{\Phi(\mathbf{w})}$, where $\mathbf{w}$ and $\mathbf{w}^*$ are the current NS configurations and the NS configurations at the NE, respectively. The value of $\psi(\mathbf{w})$ represents the efficiency of the current NS configuration $\mathbf{w}$ with respect to the NE. A high value of $\psi(\mathbf{w})$ implies that the current NS configuration is approaching the NS configuration at the NE, which implies an higher efficiency. On the contrary, the lower the value, the worse the performance.

In Fig. 3(a), we show the evolution of the potential efficiency function when $N = 50$ and the VNF Servers update their pricing policy every 30 iterations. Specifically, each VNF Server v evaluates the number $n_{v,f} = |\Gamma^f(\mathbf{w}, v)|$ of served NSBs under the current NS configuration $\mathbf{w}$, and the price $p_{v,f}$ is updated according to the stochastic-based pricing mechanism in (10). It is shown that the proposed distributed service chaining mechanism adapts to the new pricing policy by rapidly converging towards the NE in few tens of iterations. In more detail, the convergence is faster when higher values of the weigh γ are considered.

Finally, in Fig. 3(b), we plot the PoA of the proposed solution for the distributed service chaining composition problem as a function of the γ parameter for different values of V and β, when the number of NSBs is $N = 10$. It is worth noting that the price of anarchy of the obtained solution is PoA ≈ 1, that is, the obtained solution is near-optimal. Fig. 3(b) shows that the PoA increases when both the values of the weigh γ and the number V of VNF Servers increase. Instead, the PoA increases when smaller values of the weigh β (dashed lines) are considered.

In the following we present a regime analysis of the proposed architecture when $m \to +\infty$ and $\sigma_v(m) \to +\infty$. To show the effectiveness of the proposed pricing mechanism, we assume that the set of VNF Servers, $\mathcal{S}$, is divided into two distinct subsets, $\mathcal{S}_S$ and $\mathcal{S}_V$, where $\mathcal{S}_S$ is the subset of VNF Servers which use a static pricing policy, i.e., the VNF price $p_{v,f}$ is fixed and constant for the whole duration of the simulations. Instead, $\mathcal{S}_V$ represents the subset of VNF Servers which apply the pricing mechanism in (10). We assume that all VNF Servers in $\mathcal{S}_S$ use uniform pricing for all of their VNFs, and the price is set to $p_{v,f} = 100$ PU in the first simulation. Let $V_s = |\mathcal{S}_S|$ be the number of VNF Servers applying a static pricing policy. In Fig. 4 we show the outcome of the game when we gradually increase the value of $p_{v,f}$ for the static VNF Servers, for two different sizes of the set $\mathcal{S}_S$, i.e. $V_s = 1$ and $V_s = 3$. Fig. 4(a) shows the average VNF price for the two sets of VNF Servers in the two considered cases. As expected, the VNF price



(a) Average VNF price. (b) Average percentage of served flows. (c) Average profit $\Pi_v^{(\mathcal{S})}(\mathbf{p}_v)$.

Fig. 4. Analysis at the steady-state.

linearly increases for the static VNF Servers, because imposed as input, while the price calculated by the adaptive price mechanism in (10) increases as well, but becomes constant after a given value of the price imposed by the static VNF Servers in $\mathcal{S}_S$. Fig. 4(b) presents the average number of flows using the static-price and the adaptive-price VNF Servers. As expected, with increasing price of the static VNF Servers, the number of served flows decreases, while the number of flows served by the adaptive VNF Servers increases, and this occurs for both the considered cases.

Finally, Fig. 4(c) depicts the average profit gained by each VNF Server, expressed in PU. Observing the figure, we can note that the profit of the adaptive VNF Servers increases, while the profit of the static VNF Servers present a maximum that depends on the number of these servers.

5. Conclusions and Future Work

The target of this paper is to propose a road to transform the NFV market of the SDN/NFV ecosystem into a marketplace where customers of the TO can participate as third-party sellers by offering VNFaaS. System scalability is also increased in terms of management by applying a distributed solution to the service chain composition problem, by charging each TO customer in finding autonomously the best service chain of VNF instances that accommodate its individual requirements. A regime analysis has also been carried out to evaluate the impact of the price choice of some VNF Servers over the performance of the whole system. As future work, specification of the game among VNF Servers and between NSBs will be provided.

References

1. S. Sezer, S. Scott-Hayward, and S. Rao, "Are we ready for sdn? implementation challenges for software-defined networks," *IEEE Communications Magazine, Vol. 51, Issue 7, July 2013.*
2. T. Nadeau and P. Quinn, "Problem Statement for Service Function Chaining," RFC 7498, Nov. 2015.
3. T. Lukovszki, M. Rost, and S. Schmid, "It?s a match! near-optimal and incremental middlebox deployment," *ACM SIGCOMM Computer Communication Review, Vol. 46 Issue 1, January 2016.*
4. B. Batouche, Y. Naudet, and F. Guinand, "Semantic web services composition optimized by multi-objective evolutionary algorithms," in *Internet and Web Applications and Services (ICIW), 2010 Fifth International Conference on*, May 2010, pp. 180–185.
5. G. Canfora, M. Di Penta, R. Esposito, and M. L. Villani, "An approach for QoS-aware service composition based on genetic algorithms," in *Proceedings of the 7th annual conference on Genetic and evolutionary computation*. ACM, 2005, pp. 1069–1075.
6. S. Sahhaf, W. Tavernier, M. Rost, S. Schmid, D. Colle, M. Pickavet, and P. Demeester, "Network service chaining with optimized network function embedding supporting service decompositions," *Computer Networks*, vol. 93, pp. 492–505, 2015.
7. P. Wang, J. Lan, X. Zhang, Y. Hu, and S. Chen, "Dynamic function composition for network service chain: Model and optimization," *Computer Networks*, vol. 92, pp. 408–418, 2015.
8. S. D'Oro, L. Galluccio, S. Palazzo, and G. Schembra, "A marketplace as a scalable solution to the orchestration problem in sdn/nfv networks," in *IEEE Conference on Network Softwarization (NetSoft) 2017, Bologna, Italy, July 3-7, 2017.*
9. G. Xilouris, E. Trouva, F. Lobillo, J. Soares, J. Carapinha, M. J. McGrath, G. Gardikis, P. Paglierani, E. Pallis, L. Zuccaro *et al.*, "T-nova: A marketplace for virtualized network functions," in *Networks and Communications (EuCNC), 2014 European Conference on*. IEEE, 2014.
10. ETSI, "Gs nfv 003, v1.2.1 (2014-12): Network functions virtualisation (nfv); terminology for main concepts in nfv," *http://www.etsi.org/deliver/etsi_gs/NFV/001_099/003/01.02.01_60/gs_NFV003v010201p.pdf.*
11. Network Functions Virtualisation Introductory White Paper, White paper, Oct 2012.
12. ETSI NFV GS, Network Function Virtualization (NFV) Management and Orchestration, NFV-MAN 001 v0.8.1, Nov 2014.
13. R. W. Rosenthal, "A class of games possessing pure-strategy nash equilibria," *International Journal of Game Theory*, vol. 2, no. 1, pp. 65–67, 1973.

14. S. D'Oro, L. Galluccio, S. Palazzo, and G. Schembra, "Exploiting congestion games to achieve distributed service chaining in NFV networks." *IEEE Journal on Selected Areas in Communications, Vol. 35, Issue 2, pp. 407-420, 2017.*
15. G. Faraci and G. Schembra, "An analytical model to design and manage a green SDN/NFV CPE node," *Network and Service Management, IEEE Transactions on*, vol. 12, no. 3, pp. 435–450, Sept 2015.