On Pricing of 5G Services

Abstract—IT and telco providers are preparing for the era of 5G; in terms of technology, the driving force is virtualization, both for computing and networking. The 5G services will be superior than today’s online services not only in technological aspects, but also from an economic and business perspective: fast service creation, effective utilization of resources, dynamic adaption to actual demand are all direct benefits of the virtualized infrastructure. In this paper we study the economic interactions between 5G resource providers and customers: we formalize how resources should be priced and selected for being booked. In particular we show that usage-based pricing is an income-maximizing scheme for providers, and we derive the problem the customers need to solve for cost-optimizing service deployment.

I. INTRODUCTION

5G is a key driver of a “high-sounding” revolution in the networking domain. Researchers, network operators, service providers and all other stakeholders face extreme challenges to meet the requirements: the challenges stem from both technological and economic questions and open issues. Meanwhile, virtualization has significantly changed the IT realm and it is reshaping the networking ecosystem. On the one hand, it is an essential enabler of many future services, and sustainable and cost-effective network operation. On the other hand, it is a root cause of challenges we need to address in the near future.

Cloud services such as, Infrastructure as a Service (IaaS), Platform as a Service (PaaS) or Software as a Service (SaaS), are built on compute and storage virtualization techniques and they are evident results of recent years’ advancements. In networking, Network Function Virtualization (NFV) is the main driver of changes: the main goal of NFV is the softwareization of the data plane in order to transform traditional network services into software-based solutions run by cheap commodity servers. Together with Software Defined Networking (SDN), which addresses the softwareization of the control plane, they establish the basis of future networks.

Future 5G services such as, coordinated remote driving, remote surgery or Tactile Internet with its 1ms round-trip latency bound, pose extreme requirements not only on the networks but on the IT platforms. Moreover, 5G brings up the need of joint control and joint virtualization of IT and network resources. For example, the immensely fast and flexible service creation on top of shared resources requires special coordination and control workflows to be elaborated. Typical network services are realized by Service Function Chains (SFC) which can span not only over multiple domains, but over multiple operators, as well. In the novel 5G ecosystem, cost-effective, economical resource usage by sharing, and a geographically wide reach of customers are envisioned. Here, the customers are not (only) end-users, but they are Telcos, Over-The-Top (OTT) service providers and enterprise customers that deploy a service over infrastructure providers’ resources, hence becoming the primary customers of 5G infrastructure providers.

In spite of emerging NFV architectures, ongoing research projects and standardization activities targeting orchestration of compute and network resources, generally only technological aspects are tackled. We argue that in the envisaged 5G ecosystem pricing and related business aspects have significant importance and should be considered together with the technological parameters. The main motivation of our work is the fact that in this complex setup management, orchestration and pricing in particular are of the utmost importance. The price of resources must be a driving factor of the orchestration decisions as the deployed services span over multiple providers’ domains that come at various costs.

The contribution of this paper is fourfold: i) building on related work results, we define a resource usage-based pricing scheme for the 5G infrastructure providers; ii) we present a graph model of chaining virtual compute and network resources in a multi-operator and multi-domain setup; iii) we propose an integer linear program for customers to compute the set of providers that offer their resources for building a given service at the lowest cost; iv) we show a quantitative analysis of the providers’ income for cases in which they diverge from the pricing scheme that we propose.

The paper is organized as follows. In Sec. II we give an overview of the state-of-the-art to show how poorly the question of pricing is addressed, and we argue for a resource usage-based pricing scheme. In Sec. III we describe the resource orchestration model we propose and provide the algorithmic solution for customers to minimize their costs spent on 5G resources. In Sec. IV we show with simulation results why 5G providers should follow the pricing scheme we propose. We conclude our work presented in this paper in Sec. V.

II. BACKGROUND OF SERVICE PRICING

We present an overview of pricing aspects of services over compute and network infrastructures. First we summarize how little current management and orchestration (MANO) solutions tackle pricing of resources, functions and services. Then we
list the main orchestration engines. Last, we argue how well-suited dynamic pricing is for multi-domain SFC pricing.

A. Pricing in Service Definitions

Standardization bodies and industrial consortia work in parallel on MANO frameworks for cloud and NFV, but pricing aspects do not receive any attention, however there is clearly a strong relation between orchestration and pricing. Furthermore, not only the provider selection and function deployment is greatly affected by prices, but also the management of services, particularly through scaling and SLA enforcement.

First let us review the current status of service and function descriptors, as those are the front line means of driving MANO processes. Two main approaches co-exist for the description of services: the Network Service Descriptor [1] from ETSI and TOSCA [2] templates from OASIS. Although ETSI’s NFV Industry Specification Group completed its first work phase in 2014 and one of the resulted specifications that covers management and orchestration [3] contains a todo note about adding pricing info later, even the latest publicly available specifications lack such details [1], [4]. The industry considers TOSCA as a de-facto standard to define distributed cloud services: TOSCA YAML files describe the service topology as a set of nodes, relationships, dependencies, instantiation and configuration settings, monitoring, and maintenance, but no pricing field can be currently found there.

In both cases, costs can be defined to a resource or capability, but these come as abstracted values, not necessarily related to money, but derived to express affinity or power consumption costs. Although the definition of a service given in either a NSD or a TOSCA template is generic enough to include prices as attributes of nodes. Then an upper business layer can drive this attribute, in alignment with high level goals and commercial aspects, which later on is translated into the composition of the service templates to be implemented by the service layer. A possible descriptor for this upper layer is the language called Business Process Model and Notation; an initial interface connecting business processes with TOSCA is described in [5], but without direct reference to pricing models.

B. Pricing Aspects in Orchestration Engines

Consequently the main available open-source MANO engines do not yet tackle pricing in their orchestration logic. We consider the following as the main ones. Open Source MANO (OSM) delivers an open-source stack closely aligned with the ETSI NFV reference architecture [6], building on RIFT.io component Launchpad (RIFT.ware is a commercial distribution of OSM [7] from the same company) and JuJu [8], Canonical’s open source application modeling tool that deploys, configures, scales and operates the software on public and private clouds. OPEN-O [9], another major project, enables end-to-end service agility across multiple domains using unified platform for NFV and SDN orchestration. Third, Cloudify was introduced as a pure cloud orchestration solution, but the platform was further expanded to include NFV-related use cases, and the Cloudify Telecom Edition emerged [10]. Other MANO projects such as OpenBaton [11], T-NOVA/TeNor [12] and SONATA [13] have their codebases available in public repos, but while industry support, involvement of external contributors and further sustainability might be a challenging for these projects in the future [14], only traces of pricing aspects can be seen. T-NOVA’s Marketplace solution allows providers to determine billing details for their services: pay-as-you-go option can be selected, price per period and service setup cost can be given in selected currencies which is then written into the NSD. The orchestrator of the UNIFY project [15] does not tackle pricing, but optimizes on balanced usage of resources, both in terms of compute and network [16]. We show that including prices in the decision, this is the outcome the orchestration will achieve implicitly.

C. Pricing of IT Resources and Services

The main reason for the lack of pricing aspects within these MANO frameworks is that they mainly target a single administrative domain, and therefore the end-user price is not directly linked to the cost of resources that are used for provisioning the service. Most infrastructure providers today apply flat or pay-as-you-go pricing schemes with fixed unit prices in the latter case. Then the price competition is defined by these prices, and the internal costs of the provider once selected to deploy a service is minimized internally. However, the provider’s income is greatly affected by the dynamics of demand, by elasticity, i.e., scaling up/down the service.

In principle there are several ways to deal with this dynamics: either by creating markets that can generate the values of the parameters of the pricing schemes or by creating pricing schemes and market mechanisms where prices are set dynamically according to multiple factors (e.g., utilization). An example of the former is a spot market where a market price for a certain resource or service is generated given the intersection point of demand and supply: this approach is taken in Amazon EC2 Spot Instances [17], in ride-sharing systems like Uber [18], etc. The latter, dynamic pricing of services is most useful when two product characteristics co-exist. First, the product expires at a point in time, like hotel rooms, airline flights, generated electricity, or time-dated perishable ("sell before") products. Second, capacity is fixed well in advance and can be augmented only at a relatively high marginal cost. These characteristics create the potential for very large swings in the opportunity cost of sale, because the opportunity cost of sale is a potential foregone subsequent sale. The value of a unit in a shortage situation is the highest value of an unserved customer. Forecasting this value given current sales and available capacity represents dynamic pricing [19].

An example for this usage-based dynamic pricing scheme applied for NFCs is presented in [20]: based on historic data about service request arrival rate (modeled as a Markov process), about the amount and duration of resource allocations, about prices and resource usage levels, the authors derive the prices, based on the actual utilization, that maximizes the potential income of the provider. Going forward, in this paper we target a multi-provider setup, where a service is
implemented on the federation of multiple providers. Total cost is based on the prices of booked resources of participating providers: prices are added up throughout the chain from all the operators involved. We propose that the price at a given point in time of a given amount of resource booked for a given duration in the future at a given provider should vary according to the current level of booked resource utilization for that duration at that provider. This proposition builds on the findings in [20]: as booked utilization increases, providers up their prices in order to reserve capacity for later arriving customers with higher utility in order to increase their income.

III. RESOURCE MODEL FOR SERVICE CREATION

In order to grasp the complexity of the multi-provider setting we build a price model of resources and we give an algorithm that finds the resource set for the customer that can host its service at the lowest cost possible.

The actors of the ecosystem can be grouped into 3 categories. We model the customers by the Service Access Points (SAP) where they are ready to consume the service: this might be a geographic point or region, an IP address or an IP address range, even a whole autonomous system. Network service providers is the second group that we denote by NWs which refers to access and transit networks. An access network can be either wired or wireless, either a virtual operator or one with a physical network; a transit network can be a Tier1 or a Tier2 provider, even a Tier3 one with direct DC connectivity. And this brings us to the third group of actors, the Data Center Tier2 provider, even a Tier3 one with direct DC connectivity.

First we summarize the notations we use throughout this paper and present the graph model of the ecosystem, then we define the SFC orchestration problem of finding the cheapest possibility for the customer to deploy its service as a minimum-cost flow problem, finally we formulate the problem as an Integer Linear Program.

A. Graph Model and Notations

Let $\mathcal{A}$ denote the set of SAPs, $\mathcal{N}$ the set of NWs and $\mathcal{D}$ the set of DCs. We define a directed graph $G = (V, E)$ as follows, where $V$ denotes the set of nodes, and $E$ the set of arcs. There is a node assigned to each SAP, each NW and each DC in $G$, thus the total number of nodes is $|V| = |\mathcal{A}| + |\mathcal{N}| + |\mathcal{D}|$. We add the following arcs (see Fig. 1):

- Connect SAP to NW: each customer’s online service is reachable at one or more network providers;
- Interconnect NW nodes: connect the transit and access networks which have direct links, representing direct peering, connectivity at Internet Exchange Points, etc.;
- Connect NW to DC: each DC is connected to the Internet via at least one network provider.

Note, $G$ has no arcs between the nodes in $\mathcal{A}$, and similarly it has no arcs between the nodes of $\mathcal{D}$ (see Fig. 1).

A given service request $s \in \mathcal{S}$ is defined by a SAP, denoted by $a_s \in \mathcal{A}$, the maximum price $p_s$, the customer is willing to pay, a delay constraint $d_s$ that gives an upper limit on the additive network latencies the network providers guarantee, and the required resource amount $c_s$. If a service chain has multiple SAPs, in our model it can be replaced by multiple service chains each with single SAP. In our model, we assume that resources have a general unit, which is either a measure of computation required to run the service in DCs, or corresponds to consumed network bandwidth at NWs. This also implies that any DC can serve any service, i.e., the computation never requires any special hardware.

On the providers’ side we use upper-case letters. $C$ is used to denote their capacity, which can be either a network capacity $C_n$ for $n \in \mathcal{N}$, or compute capacity in a data center $C_d$ for $d \in \mathcal{D}$, and inline with the above assumption, we suppose that there is a fixed relation between the network and the compute requirements of all services. We introduce $L$ for the supposedly constant latency/delay a network provider guarantees, e.g., $L_n$. We assume zero delay in data centers, i.e., $L_d = 0$, $\forall d \in \mathcal{D}$. As pricing, we use the notations $U$ and $P$ for the current usage level and the current price of resources for both network and data center providers. As motivated in Sec. II, we assume that there is functional relation between usage and price: $P_k = f(U_k)$, $\forall k \in \mathcal{N} \cup \mathcal{D}$. The price is zero for SAP nodes, i.e., $P_a = 0$, $\forall a \in \mathcal{A}$, and those nodes do not raise any technical obstacles, i.e., $C_a = \infty$, $L_a = 0$, $\forall a \in \mathcal{A}$.

B. Service Chain as a Flow

Our goal is to provide resources for a service chain initiated from the customer’s SAPs. Typically, in each service chain there is a single SAP, which is connected to a single DC through the graph of NWs. So for deploying a given service, the customer has to select a DC and a path in the network to reach it (see Fig. 1): technically this is going to be a path in $G$ between a node in $\mathcal{A}$ to a node in $\mathcal{D}$.

Generally however, a service chain may be connected to multiple DCs if the capacity of one DC is not enough or it is too expensive. If the SAP is served by multiple DCs we may face an additional cost of synchronizing the states of the service between DCs, which we neglect here. These imply that the service chain $s \in \mathcal{S}$ can be modeled as a flow starting from its SAP $a_s \in \mathcal{A}$, with capacity $c_s$, and terminating in DCs. Moreover, a service chain may have multiple SAPs, with a given capacity requirement starting at each one of those SAPs.
In case of multiple SAPs, we treat the service as multiple flows calculated separately.

To be inline with network flow theory, first we need to redefine the price, delay and capacity values, but this time on arcs instead of on nodes, denoted by \( p_e \), \( l_e \) and \( c_e \) respectively for arc \( e \in E \), as follows:

\[
p_{u \rightarrow v} = P_v, \quad l_{u \rightarrow v} = L_v \quad \text{and} \quad c_{u \rightarrow v} = C_v .
\]

One can verify that a simple path \( T \) (without loops), has the same cost in both models, i.e., with price defined on network nodes or on arcs, formally \( \sum_{e \in T} p_e = \sum_{v \in T} P_v \). Note that as a result of cost optimization, the obtained flow corresponding to a service chain has no loops either, thus the above can be generalized to flows as well. This holds for delay too, which is also an additive metric. Similarly, one can verify that a flow that meets the capacity constraints of the nodes, also meets the capacity constraints of the links, and vice versa.

**Claim 1:** Any loopless flow has the same cost, delay and meets the capacity constraints equally in both models: with the price, delay and capacity defined on nodes or on arcs.

**C. Integer Linear Program (ILP) for SFC Orchestration**

We conjecture that the SFC orchestration problem is NP-hard; thus, we will formulate it as an ILP. Let \( e \) be an arc of the graph, and let \( x_{s,e} \) denote the amount of flow allocated for a service request \( s \). Based on the above the total price the customer pays for service \( s \), i.e., the target function to minimize, is:

\[
\sum_{e \in E} p_e \cdot x_{s,e} ,
\]

as both network and data center costs add up to the final bill. The customer budget raises a constraint to this cost:

\[
\sum_{e \in E} p_e \cdot x_{s,e} \leq p_s .
\]

Next we define the flow constraints. First, we need to ensure there are \( c_s \) flow originated at the SAP of the service chain:

\[
\sum_{a_s \rightarrow v \in E} x_{a_s \rightarrow v} \geq c_s \quad \forall s \in \mathcal{S} .
\]

Second, in each NW we have flow conservation constraints:

\[
\sum_{i \in E} x_{s,i \rightarrow v} - \sum_{v \in E} x_{s,v \rightarrow j} = 0 \quad \forall v \in \mathcal{N} .
\]

Note that we do not need to deal with DC nodes as they are always sink of the flows. Third, to consider the capacity constraints of each arc, we add the following constraints:

\[
\sum_{s \in \mathcal{S}} x_{s,e} \leq c_e \quad \forall e \in E .
\]

Finally, we address delay constraints on the NWs for each service chain \( s \). To do so, we need to add a binary working variable \( y_{s,e} \) for each arc \( e \in E \), and a non-negative real variable \( z_{s,v} \) for each node \( v \). Variable \( y_{s,e} \) indicates that there is a flow on link \( e \), formally

\[
y_{s,e} = \begin{cases} 
1 & \text{if service chain } s \text{ uses arc } e , \\
0 & \text{otherwise.}
\end{cases}
\]

This can be ensured by the following equations:

\[
x_{s,e} \leq c_{max} \cdot y_{s,e} , \quad y_{s,e} \in \{0,1\} \quad \forall e \in E, s \in \mathcal{S},
\]

where \( c_{max} \) is the maximum capacity value for each node, i.e., \( c_{max} = \max_{n \in \mathcal{V}} C_n \).

Variable \( z_{s,v} \) represents the delay of the service chain \( s \) at node \( v \) from SAP \( a_s \). It should not be larger than \( d_s \):

\[
0 \leq z_{s,v} \leq d_s \quad \forall v \in V, s \in \mathcal{S}.
\]

Last but not least, we have the following constraints:

\[
(l_{u \rightarrow v} + d_s) \cdot y_{s,u \rightarrow v} - d_s + z_{s,v} \leq z_{s,u} \quad \forall u \rightarrow v \in E, s \in \mathcal{S},
\]

which ensure that if arc \( u \rightarrow v \) is traversed by the service chain, i.e., \( y_{s,u \rightarrow v} = 1 \), then the delay at node \( v \) for service chain \( s \) is (at least) \( z_{s,v} = z_{s,u} + l_{u \rightarrow v} \), otherwise the constraint on the delay is relaxed (because the difference between variables \( z_{s,v} \) and \( z_{s,u} \) is at most \( d_s \)).

**IV. NUMERICAL ANALYSIS**

In this section we turn back to the pricing scheme of the providers, and with simulations we show the motivation behind the usage-based dynamic pricing. In the 5G market, the NWs compete with each other in order to make sales and increase their income, but they also need to take into account that subsequently arriving customers might pay more for the resource they sell, which is on the other hand perishable, i.e., resource utilization expires at the point in time when the booked period ends.

**A. Simulation Settings**

**Topology.** We simulate an environment of 200 SAPs, 60 NWs (10 transit and 50 edge networks) and 200 DCs being interconnected in hierarchical topology of three tiers. All the transit NWs belong to the first tier and they are connected in a full-mesh, while each serves a number of edge NWs of the second tier. In particular, the edge NWs are assigned to the transit NWs in a uniformly distributed random way. Finally, the third tier consists of DCs and SAPs, who are also uniformly distributed to the available edge NWs via connections. We assume that an edge NW can be connected to more than one DCs or multiple SAPs, but we omit the case where an edge NW hosts SAPs and DCs simultaneously. The minimum cost solution for the SFC in this case would be mostly trivial since SAPs would always choose local DCs.

**Resource capacity and network delay.** We assume that the compute resource capacity is the same at all DCs and it is normalized to 1. The total network resource capacity of an edge NW is determined by the number of DCs or SAPs connected to it, i.e., we add 1 unit of capacity per link. For the transit NWs we follow the same approach, but we double the capacity to capture the fact that congestion rarely happens in the core. A fixed delay is assigned to each NW with uniformly distributed random values in the range of 10−20 for the edge and 2−5 for the transit NWs.

**Service requests.** We generate multiple request series of services with various lengths, ranging from 6000 up 24000
requests. For each request we select a source SAP in a uniformly distributed way. A request is assigned with a randomly (with uniform probability distribution) selected: (i) capacity (network and compute resource) in the range 0.01 – 0.02, (ii) maximum delay in the range 60 – 35 and (iii) maximum price that the customer is willing to pay in the range 0.1 – 0.2. The selected range of values for the capacity denotes that a DC can serve 50-100 requests; the range of values for the maximum delay reflects the average length of a service chain, i.e. 4, and is adapted to the selected range of values for the per link delay. Finally, the range of values for the maximum price is set to be in line with the normalized DC capacity values, as follows.

**Pricing.** Each provider charges for each service it serves and the price is based on the current utilization level of its infrastructure. We assume that all DCs or NWs use the same linearly increasing pricing function, and the maximum price they can charge at 100% utilization is set to 1. The total price that a customer pays is the sum of the prices requested by the different providers in the service chain.

**Service composition.** We assume that the service requests are served in a first-come-first-served fashion. We calculate the minimum cost path for each request solving the ILP in Sec. III and if all criteria are satisfied, we book resources for the service on the resource graph. After each booking, the involved providers re-set their prices to the actual booked utilization level of their infrastructure. We investigate the case where a provider deviates from this standard pricing policy setting its price higher or lower (±Δ).

### B. Results

We mitigate the effects of randomness in our results by running simulations for 10 different topologies and 10 service request series. The results reveal that the combination of the usage-based pricing and minimum cost path selection leads to load balancing (Table I) with high service completion (Fig. 2).

Fig. 2 shows the number of successful and unsuccessful service compositions as the service request series gets longer, i.e., more services need to be deployed. We can observe that as the number of requests increases and the resource utilization rises to a high level, i.e., over 90%, the number of unsuccessful service composition attempts starts to increase. This growth is an expected behavior since a service request can be rejected for three different reasons: (i) there is no path that satisfies the minimum delay guarantee, (ii) available paths are more expensive than what the customer is willing to pay, (iii) there is a resource scarcity on all available paths. At low utilization levels and thus with low prices, a service request might be rejected most probably for demanding extremely low delay guarantees. As the utilization increases, so does the price, therefore after a certain level some requests can be rejected because of low willingness to pay. Finally, as the system reaches its capacity limits, only requests that combine high willingness to pay and a demand of relatively low capacity can be served. Table II maps the number of unsuccessful service compositions of Fig. 2 to these three reasons. Note that a request may be rejected for multiple reasons on any candidate path. The results show that usage-based pricing is a useful mechanism to avoid extreme infrastructure utilization.

Fig. 3 depicts the total income of providers as the length of the service request series increases. We can observe that the income starts increasing almost linearly with the number of requests, but it becomes concave as the utilization level grows further since the successful composition ratio decreases.

Finally, we investigate the case where a provider deviates from the usage-based pricing and intentionally sets its price higher/lower than it should set to. We randomly select a provider and we run again the experiments with it following either a conservative (lower) or an aggressive (higher) pricing.
In particular, after setting its usage-based price, we increase (or respectively decrease) the value by $\delta = 0.2$. Fig. 4 depicts the income of the selected provider for the three scenarios while Fig. 3 shows the total income for all providers. The results reveal that a provider who follows a conservative pricing policy, attracts customers and thus becomes over-utilized compared to others. This may bring extra profit when the resource supply is high compared to the demand, but as the demand increases this policy leads to profit loss (e.g. for service series length $> 18 \times 10^3$ in Fig. 4). On the other hand, being aggressive in pricing leads to under-utilization, especially when supply is high compared to demand, however, when the availability of resources is low, an aggressive provider will still have resources to supply to the high utility customers and therefore will generate profit. Both of these approaches may have an advantage in very extreme cases, but on average the usage-based pricing generates higher income. Note that in Fig. 4, the results for each different series length is an average of multiple runs and with a random selection of the deviating provider each time. Due to this randomness, we do not expect the plots to be strictly monotone.

![Fig. 3. Total income of providers for variable lengths of service request series.](image)

![Fig. 4. Income of the selected provider for three different pricing policies.](image)

V. Conclusion

With this work we tackled the questions around resource pricing in the envisioned context of 5G online services. Major efforts are being made towards this vision: developing IT virtualization techniques and NFV advancements pave the way to the resource-optimized and elastic infrastructure lying below the future 5G services for which even the borders between providers are torn down. With this pioneer work we show, backed with a numerical analysis, how these providers will probably price their resources, and in turn we build a model and we provide a formulation for the problem that customers will face when selecting the right set of providers’ resources for deploying their services.

The model and analysis proposed in this paper are meant to be the first in their kind, i.e., pricing of multi-operator SFCs. We intend to continue the work by alleviating our assumptions made in our model, and by fabricating heuristics that make SFC orchestration fast and close-to-optimal even in the more complex setting. Furthermore, instead of numerical simulations, we plan to perform a game theoretical analysis of the pricing schemes applied by the operators, in particular studying the interplay between the connectivity graph of the providers and their income-maximizing pricing strategies.

ACKNOWLEDGMENT

This work has been performed in H2020-ICT-2014 project 5GEx (www.5gex.eu, Grant Agreement no. 671636), partially funded by the European Commission. The research was partially supported by the Hungarian Scientific Research Fund (grant No. OTKA 108947). Laszló Toka was partially supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences.

REFERENCES

[1] “Network Service Templates Specification,” ETSI GS NFV-IFA 014, Tech. Rep., 10 2016.
[2] “TOSCA Simple Profile for NFV,” OASIS Topology and Orchestration Specification for Cloud Applications (TOSCA) TC, Tech. Rep., 03 2016.
[3] “Management and Orchestration,” ETSI GS NFV-MAN 001, Tech. Rep., 12 2014.
[4] “VNF Packaging Specification,” ETSI GS NFV-IFA 011, Tech. Rep., 10 2016.
[5] O. Kopp et al., “BPMN4TOSCA: A Domain-Specific Language to Model Management Plans for Composite Applications,” in Business Process Model and Notation, ser. Lecture Notes in Business Information Processing, vol. 125. Springer, 2012, pp. 38–52.
[6] Open Source MANO. [Online]. Available: https://osm.etsi.org/
[7] RIFT.ware. [Online]. Available: https://rifto.com/riftware/
[8] Juju. [Online]. Available: https://jujucharms.com/
[9] OPEN-O. [Online]. Available: https://www.open-o.org/
[10] Cloudify. [Online]. Available: http://getcloudify.org/
[11] OpenBaton. [Online]. Available: https://openbaton.github.io/
[12] T-NOVA. [Online]. Available: http://www.t-nova.eu/open-source/
[13] SONATA. [Online]. Available: http://www.sonata-nfv.eu/
[14] D. Andrusiak et al. (2017) What is the best NFV Orchestration platform? A review of OSM, Open-O, CORD, and Cloudify. [Online]. Available: https://www.mirantis.com/blog/which-nfv-orchestration-platform-best-review-osm-open-o-cord-cloudify/
[15] B. Németh et al., “Efficient service graph embedding: A practical approach,” in IEEE NFD-SDN, 2016.
[16] B. Sonkoly et al., “UNIFYing Cloud and Carrier Network Resources: An Architectural View,” in IEEE GLOBECOM, 2015.
[17] Amazon ec2 spot instances. [Online]. Available: https://aws.amazon.com/ec2/spot/
[18] K. Chen et al., “Dynamic pricing in a labor market: Surge pricing and flexible work on the uber platform,” Mimeo, UCLA, Tech. Rep., 2015.
[19] R. P. McAfee et al., “Dynamic pricing in the airline industry,” Economics and Information Systems, 2006.
[20] B. Naudts et al., “A dynamic pricing algorithm for a network of virtual resources: Revenue management for infrastructure providers,” International Journal of Network Management, vol. 27, no. 2, 2017.