Economics of Multi-Operator Network Slicing

George Darzanos, Iordanis Koutsopoulos, Katia Papakonstantinopoulou, George D. Stamoulis
Athens University of Economics and Business, AUEB, Athens, Greece
{ntarzanos, jordan, katia, gstamoul}@aueb.gr

***Abstract*—Network slicing allows Mobile Network Operators (MNOs) to partition their physical infrastructure into multiple virtual logical networks, enabling the simultaneous servicing of applications with diverse Quality of Service characteristics. In this paper, we introduce and evaluate economic models and policies for the provisioning, across multiple MNOs, of network slice services to Application Providers. We introduce a Network-Slice-as-a-Service model that maps the service offered by a network slice to requirements on virtualized resources. The placement of virtualized resources over the physical infrastructure of MNOs is determined by an embedding problem, formulated as a Mixed Integer Program. We investigate the embedding under: (i) centralized approaches, where a central Broker determines the embedding for all network slice requests, and (ii) peer-to-peer approaches, where each MNO determines the embedding for the sub-set of network slice requests coming from its own customers. We introduce policies for cooperative modes, with the objective of total profit maximization, and for "coopetitive" (cooperative competition) modes, where MNOs aim to maximize their individual profits. The numerical results reveal that MNOs can maximize their aggregate and individual profits under any approach or mode, if they comply with the proposed policies.**

I. Introduction

Network slicing is a network virtualization paradigm that allows Mobile Network Operators (MNOs) to manage their networks in an agile and automated way. The infrastructure of MNOs is partitioned into multiple virtual logical networks (i.e., network slices), each guaranteeing a certain level of Quality of Service (QoS). Based on their QoS characteristics, network slices are classified into three types, namely the enhanced Mobile Broadband (eMBB), ultra Reliable Low Latency Communications (uRLLC) and massive Machine Type Communications (mMTC) network slices.

Building on network slicing, 5G networks promise to boost the digital transformation in sectors such as automotive, industry 4.0, media, etc., since MNOs can support the simultaneous provisioning of novel applications with diverse and stringent QoS requirements. Following the Network-Slice-as-a-Service (NSaaS) model, MNOs offer Application Providers network slices that are "tailored" to the needs of their that may server users located to different geographic regions. *Service-wise*, a network slice is a chain of interconnected Virtual Network Functions (VNFs), combined with application-specific functions, dedicated to servicing the traffic generated by the application it enables. *Resource-wise*, a network slice consists of network and computational resources that may need to be allocated across different network domains or MNOs, as Virtual Machines (VMs) and Virtual Tunnels (VTs).

Given that an MNO provisions multiple network slices, the fundamental problem is: How can an MNO determine the placement (embedding) of VMs and VTs to its physical infrastructure in a manner that maximizes its profit, while respecting the Application Providers' QoS? Different flavors of this problem have been studied in virtual network embedding [1], VNF chaining [2], and network slicing [3] literature. The embedding problem becomes challenging when resources from multiple MNOs are required for establishing a network slice. This usually happens when the traffic generated by a *source* User Equipment (UE) in a certain geographic region, has to be delivered to UEs located in remote *destination* regions that cannot be served by a single MNO. For instance, a surgeon located in UK may need to use a Virtual Reality (VR) application in order to control a robot that performs surgery in a hospital located in Italy. Then, VNFs related to VR equipment have to be deployed in UK, while VNFs related to robot control have to be deployed in Italy. Assuming that there is not a single MNO that has presence in both countries, there is a need for resources from at least two MNOs.

The multi-MNO embedding problem becomes even more challenging if we consider that: (i) MNOs are rational entities that seek profits and the competition among them may lead to inefficient resource allocation; (ii) MNOs may have incomplete knowledge about others' resource availability and QoS capabilities. Therefore, we need to establish mechanisms that incentivize and facilitate the collaboration of MNOs. The recent literature studies the economic interactions of an MNO with multiple VNF providers [4], VNF users [5] or network slice tenants [6], but not between different MNOs.

Our Contribution. We study the economics of interactions among MNOs when jointly offering network slices as a service to Application Providers (customers), under both *centralized* and *peer-to-peer* approaches. In the *centralized* approach, the embedding of all network slice requests is determined by a central Broker. In the *peer-to-peer* approach, each MNO determines the embedding of network slice requests of its own customers, through bilateral interactions with the others. We introduce policies for a *cooperative* mode (suitable for the centralized approach), where the MNOs have as common objective the *total profit* maximization. The MNOs trust the Broker to solve a Mixed Integer Program (MIP) for performing a *global* embedding of all network slice requests. We also introduce policies for a *coopetitive* (cooperative competition) mode, where MNOs aim to maximize their individual profits but they still collaborate for offering joint network slices. Therefore, multiple *local* embedding sub-problems are formulated, i.e., one per MNO. In the centralized approach, the Broker solves all sub-problems on behalf MNOs, while in the peer-to-peer approach each MNO solves its own sub-problem.

In Section II, we present the related work. In Section III, we introduce the system model. In Section IV, we study the multi-MNO network slice embedding for centralized and peer-to-peer approaches, introducing policies for both cooperative and coopetitive modes. In Section V, we present our numerical evaluation results. In Section VI, we present our conclusions.

II. Related Work

Virtual Network (VN) embedding. In [1] and [7], the problem of VN embedding over a shared infrastructure controlled by a single administrative entity is formulated as an Integer Program. In [8], a distributed VN embedding framework for multi-MNO scenarios is proposed and a pricing mechanism is introduced. However its efficiency or the potential abuse of it was not investigated. In [9], the problem of multi-MNO virtual resources provisioning is addressed, by taking advantage of max-flow/min-cut algorithms and Integer Linear Programming (ILP). However, competition and economic aspects were not considered. The authors in [10] and [11] study the problem of multi-MNO VN embedding with limited information.

VNF chaining and placement. In [2], the VNF placement and chaining problem is formulated as an ILP. A heuristic that achieves a close-to-optimal VNF allocation in terms of cost and delay is also proposed. In [12], a graph theory-based heuristic is proposed for solving the problem of determining the required number and placement of VNFs that optimize network operational costs. In [4] and [5], the authors use the game theory framework in order to propose mechanisms that achieve VNF chaining and resource allocation in a distributed way. However, in [4], the impact of VNF chaining is not captured, while [5] focuses on the case of a single MNO.

Network Slicing. In [3], an algorithm for the fair allocation of resources to multiple network slices in introduced. In [13], a polynomial time heuristic for the embedding of network slices with splittable flows is proposed. In [14], three alternative policies for resource management in 5G networks are evaluated, considering multiple QoS classes. In [6], the problem of network slicing is studied when multiple slice tenants compete over shared resources.

Some of the above works that study VN embedding, VNF chaining and network slicing are focusing either on a single MNO domain or on non-economic aspects of multi-MNO service provisioning. The works that have a flavor of economics (namely [4], [5] and [6]) do not study the multi-MNO setting. In this paper, we study the economic interactions among MNOs and we introduce policies for both centralized/peer-to-peer approaches and cooperative/coopetitive modes. Finally, our system model and problem formulation are both richer that those of the existing works since they capture all resource capacity, latency, geographic and economic aspects.

III. System Model

A. Topology and Resource Provisioning Model

Let $\mathcal{I}$ denote the set of MNOs that collectively build an interconnected virtualized infrastructure. We assume that the overall topology is modeled as a graph $G = (\mathcal{I}, E)$, where each node $i \in \mathcal{I}$ represents the abstracted internal topology of an MNO, while each edge $e_{ij} \in E$ denotes the existence of a direct interconnection between MNOs i and j (possibly through an IXP). MNOs are dispersed in $|\mathcal{L}|$ geographic regions. We assume that each MNO $i \in \mathcal{I}$ has presence at a single region $L_i \in \mathcal{L}$, while multiple MNOs may be present in a region. Figure 1 shows an instance of an abstract topology for $|\mathcal{I}| = 10$ MNOs dispersed in four geographic regions. Each MNO i maintains computational capacity of C_i (CPU cores), while the bandwidth of a network link $e_{ij} \in E$ is B_{ij}. We assume that $B_{ij} = B_{ji}$. Assuming that the network is properly dimensioned and that all network links remain busy, the traffic traversing link e_{ij} suffers a delay D_{ij}, which is assumed to be symmetric, i.e., $D_{ij} = D_{ji}$. Computational resources are provisioned within the graph nodes (i.e., MNOs) as VMs and each of them hosts a VNF. Network resources are provisioned in the form of Virtual Tunnels (VTs) of specific QoS that handle the traffic of the different network slices.

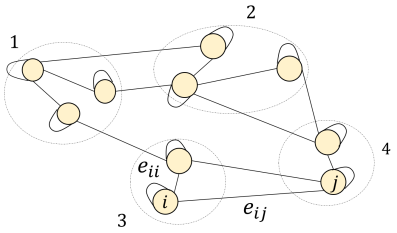


Fig. 1. Abstract topology of 10 MNOs dispersed into four geographic regions.

B. Network Slice as a Service Model

A network slice service is realized by a chain of VNFs and application-specific functions that guarantee a certain QoS. Hence, it can be abstracted as a directed acyclic graph, where each VNF is represented by a node and edges indicate how the traffic flows through the sequence of VNFs. The top part of Fig. 2 depicts such an example graph, where four VNFs form a network slice service chain. Each VNF f is hosted on a VM $\nu \in \mathcal{V}$ of adequate computational capacity, while the interconnection of VMs is enabled through an *assured quality* VT τ. The service requirements captured by the *service graph* can be mapped to virtualized resources captured by the non-directed *resource graph* at the bottom part of Fig. 2.

The amount of CPU cores allocated to VM ν is denoted by c_ν. A VT τ is characterized by a tuple (b_τ, d_τ), where b_τ is the allocated physical bandwidth, and d_τ is the guaranteed latency. Each MNO incurs a certain operational cost ($\$/sec$) for maintaining the VMs and VTs of different network slices

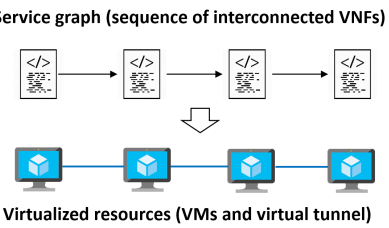


Fig. 2. Transformation of a service graph into a virtualized resource graph.

up and running. We assume that the cost of an MNO i for maintaining a VM ν active is a function that linearly increases with the capacity c_ν that has to be allocated, i.e. $\kappa_i(c_\nu) = c_\nu \kappa_i^{co}$, where κ_i^{co} is the cost of a single CPU core in MNO i. Respectively, when a VT τ of bandwidth b_τ traverses a physical link e_{ij}, it generates a cost $\kappa_{ij}(b_\tau) = b_\tau \kappa_{ij}^{bw}$.

Network slice request. Let $\mathcal{R}$ denote the set of all network slice requests. Each request $r \in \mathcal{R}$ assigns values to the following parameters of a *network slice template*:

1) Network slice type: If $\mathcal{F}$ is the set of all VNFs and $\mathcal{T}$ is the set of available network slice types (eMBB, uRLLC, etc.), the selected type $t_r \in \mathcal{T}$ determines the subset $\mathcal{F}_{t_r} \subseteq \mathcal{F}$ and sequence of VNFs that have to be deployed and the size of data packets K_{t_r} that will flow through.

2) Quality class: Set $\mathcal{Q}$ denotes the available quality classes (e.g. Standard, Premium) at which a network slice can be offered. The selected class $q_r \in \mathcal{Q}$, combined with the type t_r, determine the required throughput $B(t_r, q_r)$ and end-to-end latency $D(t_r, q_r)$ at network slice level.

3) Region of source and destination: It defines the pair $L_r = (L_r^{src}, L_r^{dst})$ of source and destination geographic regions of the service r. Note that $L_r^{src}, L_r^{dst} \in \mathcal{L}$.

4) VM placement restrictions: Denote any potential restrictions with respect to the geographic region that each VM should be deployed. A VM may have to be strictly placed to an MNO in the source (or destination) geographic region of the service. Let $\boldsymbol{\ell}_r = \{\ell_\nu\}_{\nu \in \mathcal{V}_r}$ be a set of placement restrictions for all VMs of request r. Each element $\ell_\nu \in \{-1, 0, 1\}$ determines whether VM ν should be placed in an MNO at the *source* (if $\ell_\nu = -1$) or *destination* (if $\ell_\nu = 1$) regions. If $\ell_\nu = 0$, then there is no restrictions for the respective VM and can be place to any MNO across the selected path.

5) Traffic rate: λ_r (in packets/sec) denotes the average rate of traffic that needs to be handled by network slice r.

6) Price: $\hat{p}_r$ denotes the price that the customer is willing to pay for the type and quality of the network slice r.

QoS model. It accounts for the throughput of a network slice (for all application users) and the end-to-end latency.

(i) *Throughput.* It is defined as the traffic (in packets/sec) that passes through the VT and VMs of a network slice. Considering a request r, the throughput achieved by a VT τ_r, is determined by b_{τ_r}/K_{t_r}, i.e., the amount of allocated bandwidth b_{τ_r} (which is reserved over all physical links that τ_r crosses) and data packet size K_{t_r}. The throughput achieved by a VM $\nu \in \mathcal{V}_r$ is determined by its service rate $\mu_\nu(c_\nu, f)$, which depends on the amount of allocated resources c_ν and the actual task that the hosted VNF f needs to perform:

$$\mu_\nu(c_\nu, f) = \sigma_f c_\nu \ , \tag{1}$$

where σ_f is the number of data packets per second that VNF f can process over a *unit* of computational capacity. This captures the fact that different VNFs may require different amount of resources for achieving the same throughput.

(ii) *Latency.* The end-to-end latency depends both on the VT latency and the processing delay in VMs. The VT latency is determined by the time required for a data packet to traverse the *path* of physical links over which the VT has been deployed. We use $\mathcal{P}$ to denote a small set that includes few feasible paths for each pair of MNOs. These paths can be the lowest-latency or lowest-cost ones, or even the ones that are usually selected (based on historical data). If VT τ_r is deployed over a path $\pi \in \mathcal{P}$ ($\pi \subseteq E$), the VT latency $d_{\tau_r}(\pi)$ is given by aggregating the delays of all physical links along this path. The processing delay of traffic when it passes through a VM $\nu \in \mathcal{V}_r$ depends on ν's service rate, $\mu_\nu(c_\nu, f)$. We assume that data packets, from multiple users of the application provisioned over network slice r, arrive to VMs according to a Poisson process of average rate λ_r, while the service time of each packet follows an exponential distribution with mean $1/\mu_\nu(c_\nu, f)$. Hence, the service offered by a VM can be modeled as an $M/M/1$ queueing system. Then, the processing delay in VM $\nu \in \mathcal{V}_r$ that hosts VNF $f \in \mathcal{F}_{t_r}$ is

$$d_\nu(c_\nu, f) = \frac{1}{\mu_\nu(c_\nu, f) - \lambda_r}. \tag{2}$$

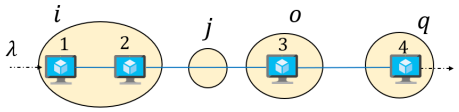


Fig. 3. Network slice embedding example (4 MNOs, 4 VMs and 1 VT).

Network slice dimensioning. The dimensioning process determines the amount of virtualized resources that will be allocated to each request $r \in \mathcal{R}$, for achieving target throughput $B(t_r, q_r)$ across all network slice elements. In particular, to achieve *throughput* $B(t_r, q_r)$ in VT τ_r, we need to set $b_{\tau_r} = B(t_r, q_r)\, K_{t_r}$, while to achieve $B(t_r, q_r)$ in VM $\nu \in \mathcal{V}_r$ that hosts a VNF $f \in \mathcal{F}_{t_r}$, we need to allocate an amount of computational resources c_ν that is given by

$$\mu_\nu(c_\nu, f) = B(t_r, q_r) \overset{(1)}{\Longrightarrow} c_\nu = \left\lceil \frac{B(t_r, q_r)}{\sigma_f} \right\rceil. \tag{3}$$

IV. Multi-Operator Network Slice Embedding

While the dimensioning a network slice is performed upon the receipt of request, the embedding process is performed periodically. In this process, the *dimensioned* network slices are embedded to the physical topology $G = (\mathcal{I}, E)$. Given a request r, the embedding determines which MNO will host each VM $\nu \in \mathcal{V}_r$ and over which physical path $\pi \in \mathcal{P}$ the VT τ_r will be placed. Figure 3 illustrates an example of a network slice embedded to the infrastructure of four MNOs.

We study embedding under both centralized and peer-to-peer approaches. In the *centralized* approach, a Broker serves as an *one-stop-shop* for all requests and performs the embedding on behalf of all MNOs. The Broker obtains global knowledge by periodically gathering information from MNOs, related to availability and cost/price of resources. Based on this information, the Broker determines which MNOs should contribute resources to each request. In the *peer-to-peer* approach, each MNO determines the embedding across multiple MNOs of requests coming from its own customers, through bilateral interactions with the other MNOs and based on knowledge obtained through an information sharing mechanism.

A. Centralized Approach Policies

*1) **Centralized Cooperative mode**:* The MNOs share the common objective of *total profit maximization* and trust the Broker to perform a *global* embedding for achieving it, by solving an instance of MIP. The resulting total revenues are distributed among the involved MNOs based on a *fair policy*.

Let $\mathbf{Y}$ and $\mathbf{X}$ denote the set of *binary decision variables* that determine the placement of all VMs and VTs. A variable $y_{\tau_r,\pi} \in \mathbf{Y}$ determines whether the VT τ_r is placed over physical path π (for $y_{\tau_r,\pi} = 1$) or not ($y_{\tau_r,\pi} = 0$). Accordingly, a variable $x_{\nu,i} \in \mathbf{X}$ determines whether VM ν is placed in MNO i (for $x_{\nu,i} = 1$) or not ($x_{\nu,i} = 0$). The Broker solves an instance of MIP that takes into account the payments that MNOs receive from Applications Providers (first term), and the cost of MNOs for deploying deploying the necessary VMs (second term) and VTs (third term):

$$\max_{\mathbf{X},\mathbf{Y}} \sum_{r\in\mathcal{R}} \Bigg[\sum_{\pi\in\mathcal{P}} y_{\tau_r,\pi}\ \hat{p}_r - \sum_{i\in\mathcal{I}} \sum_{\nu\in\mathcal{V}_r} x_{\nu,i}\kappa_i(c_\nu) - \sum_{\pi\in\mathcal{P}} y_{\tau_r,\pi} \sum_{e_{ij}\in\pi} \kappa_{ij}(b_{\tau_r}) \Bigg] \quad (4)$$

$$\text{s.t. } (6)-(11), \text{ explained below.}$$

Unique placement constraints. Constraints (5a) and (5b) ensure that each VM and VT, respectively, are deployed once.

$$\sum_{i\in\mathcal{I}} x_{\nu,i} \le 1, \quad \forall \nu \in \mathcal{V}_{r\in\mathcal{R}} \quad (5a)$$

$$\sum_{\pi\in\mathcal{P}} y_{\tau_r,\pi} \le 1, \quad \forall r \in \mathcal{R} \quad (5b)$$

Infrastructure capacity constraints. Constraint (6a) guarantees that the aggregate resources of the VMs deployed in an MNO do not exceed its available capacity. Constraint (6b) ensures that the aggregate bandwidth assigned to all VTs crossing a physical link does not exceed its capacity. Note that $z(e_{ij},\pi)$ is an indicator function that returns 1 if path π includes edge e_{ij}, otherwise it returns 0.

$$\sum_{r\in\mathcal{R}} \sum_{\nu\in\mathcal{V}_r} x_{\nu,i}\ c_\nu \le C_i, \quad \forall i \in \mathcal{I} \quad (6a)$$

$$\sum_{r\in\mathcal{R}} \sum_{\pi\in\mathcal{P}} y_{\tau_r,\pi}\ b_{\tau_r}\ z(e_{ij},\pi) \le B_{ij}, \quad \forall e_{ij} \in E \quad (6b)$$

VM placement restrictions. Constraints (7a) and (7b) ensure that the VM placement restrictions, as identified by the customer, are satisfied. When a VM ν must be deployed at the source geographic region (i.e., when $\ell_v = -1$), constraint (7a) guarantees that the VM will be only deployed to an MNO i (i.e., $x_{\nu,i}$ can take the value 1) that has presence in the source region, i.e., $L_i = L_r^{src}$. Indeed, for all MNOs that are not located in the source region, all terms in the left hand side of (7a) are non-zero except for $x_{v,i}$, which implies that $x_{v,i} = 0$. Similarly, constraint (7b) guarantees the placement restrictions for the VMs that must deployed to the destination region.

$$x_{\nu,i}\ \ell_\nu\ (1-\ell_\nu)(L_i - L_r^{src}) = 0, \forall \nu \in \mathcal{V}_{r\in\mathcal{R}}, \forall i \in \mathcal{I} \quad (7a)$$

$$x_{\nu,i}\ \ell_\nu\ (\ell_\nu+1)(L_i - L_r^{dst}) = 0, \forall \nu \in \mathcal{V}_{r\in\mathcal{R}}, \forall i \in \mathcal{I} \quad (7b)$$

TABLE I
NOTATION TABLE

Notation	Context
$\mathcal{I}$, $\mathcal{L}$, E	sets of MNOs, geographic regions, physical network links
L_i	geographic region that MNO i has presence
C_i	computational resource capacity of MNO i
e_{ij}	physical network link that interconnects MNOs i and j
B_{ij}	bandwidth of physical network link e_{ij}
π	path, a sequence of physical network links
$\mathcal{P}$	set of feasible paths between all pairs of MNOs
$z(e_{ij},\pi)$	indicator function, returns 1 if path π includes edge e_{ij}
$\mathcal{R}$, $\mathcal{F}$	sets of total requests, available VNFs
$\mathcal{T}$, $\mathcal{Q}$	sets of available network slice types, quality classes
t_r, q_r	network slice type and quality class of request r
$\mathcal{F}_{t_r}$	set of VNFs required for request r
$B(t_r,q_r)$	target throughput for request r
$D(t_r,q_r)$	target end-to-end latency for request r
L_r^{src}, L_r^{dst}	source and destination geographic regions of request r
$\mathcal{V}_r$	set of VMs required for request r
c_ν	computational resource capacity allocated to VM ν
$\mathbf{c}_r$	set that maintains capacities for all VMs of request r
ℓ_ν	deployment region restrictions for VM ν
τ_r, b_{τ_r}	VT of request r, bandwidth allocated to it
$\hat{p}_r$	the price that a the customer is willing to pay for r
$\kappa_i(c_\nu)$	cost of MNO i for hosting VM ν
$\kappa_{ij}(b_\tau)$	cost of MNO j enabling VT τ over link e_{ij}
$x_{\nu,i}$	determines if VM ν will be deployed in MNO i (for $x_{\nu,i}$ = 1) or not (for $x_{\nu,i}$ = 0)
$y_{\tau_r,\pi}$	determines if VT τ_r will be deployed over path π (for $y_{\tau_r,\pi}$ = 1) or not (for $y_{\tau_r,\pi}$ = 0)

VMs and VTs alignment constraints. Constraint (8a) ensures that a VT will be provisioned *if and only if* all VMs of the respective network slice are provisioned. Constraint (8b) guarantees that the VMs of a network slice can only be placed to MNOs that appear on the path over which the VT of this network slice is placed. Recall that $z(e_{ii},\pi)$ is an indicator function that returns 1 if path π includes edge e_{ii}, i.e., if MNO i appears on path π. Thus, a VM ν can be deployed in MNO i only if $z(e_{ii},\pi) = 1$ and $y_{\tau_r,\pi} = 1$.

$$\sum_{\pi\in\mathcal{P}} y_{\tau_r,\pi} = \sum_{\nu\in\mathcal{V}_r} \sum_{i\in\mathcal{I}} \frac{x_{\nu,i}}{|\mathcal{V}_r|}, \quad \forall r \in \mathcal{R} \quad (8a)$$

$$x_{\nu,i} \le \sum_{\pi\in\mathcal{P}} y_{\tau_r,\pi}\ z(e_{ii},\pi), \quad \forall\ (\nu \in \mathcal{V}_{r\in\mathcal{R}}, i \in \mathcal{I}) \quad (8b)$$

Latency constraint. The *end-to-end latency* of a network slice r is given by the formula below, which captures the network delay d_{τ_r} of VT τ (first term) and the processing delay in all VMs in $\mathcal{V}_r$ (second term):

$$\mathbf{D}(y_{\tau_r,\pi},\mathbf{c}_r) = \sum_{\pi\in\mathcal{P}} y_{\tau_r,\pi}\ d_{\tau_r}(\pi) + \sum_{\nu\in\mathcal{V}_r} d_\nu(c_\nu, f). \quad (9)$$

Then, constraint (10) guarantees that the target value for the end-to-end latency of each network slice is not violated.

$$\mathbf{D}(y_{\tau_r,\pi},\mathbf{c}_r) \le D(t_r,q_r), \quad \forall\ (r \in \mathcal{R},\ \pi \in \mathcal{P}) \quad (10)$$

Price constraint. Constraint (11) guarantees that the aggregate cost of MNOs for the VMs and VT of request r does not exceed the price that the customer is willing to pay.

$$\sum_{i\in\mathcal{I}} \sum_{\nu\in\mathcal{V}_r} x_{\nu,i}\ \kappa_i(c_\nu) + \sum_{\pi\in\mathcal{P}} y_{\tau_r,\pi} \sum_{e_{ij}\in\pi} \kappa_{ij}(b_{\tau_r}) \le \hat{p}_r, \quad \forall r \in \mathcal{R} \quad (11)$$

The above problem is a MIP instance, which is NP-complete and can be solved by standard optimization software tools. However, its solution does not determine how the payments coming from Application Providers will be distributed to the involved MNOs. Inspired by Nash-bargaining [15], we then propose an approach to fairly distribute these payments.

Revenue sharing rule. All MNOs contributing to a network slice will cover their costs, while the net benefit will be shared proportionally to the level of their contribution. Having the optimal placement of all VMs $\mathbf{X}^*$ and VTs $\mathbf{Y}^*$, the cost of MNO i for contributing resources to request r is given by

$$K_{i,r}(\mathbf{X}^*, \mathbf{Y}^*) = \sum_{\nu \in \mathcal{V}_r} x_{\nu,i}\ \kappa_i(c_\nu) + \sum_{\pi \in \mathcal{P}} y_{\tau_r,\pi} \sum_{j \in \mathcal{I}} z(e_{ji}, \pi)\ \kappa_{ji}(b_{\tau_r}), \tag{12}$$

while the net benefit that needs to be shared is given by

$$S_r(\mathbf{X}^*, \mathbf{Y}^*) = \sum_{\pi \in \mathcal{P}} y_{\tau_r,\pi}\ \hat{p}_r - \sum_{i \in \mathcal{I}} K_{i,r}(\mathbf{X}^*, \mathbf{Y}^*). \tag{13}$$

Then, the compensation of MNO i for request r is

$$\hat{p}_{i,r}(\mathbf{X}^*, \mathbf{Y}^*) = K_{i,r}(\mathbf{X}^*, \mathbf{Y}^*) + \frac{K_{i,r}(\mathbf{X}^*, \mathbf{Y}^*)}{\sum_{j \in \mathcal{I}} K_{j,r}(\mathbf{X}^*, \mathbf{Y}^*)} S_r(\mathbf{X}^*, \mathbf{Y}^*). \tag{14}$$

*2) **Centralized Coopetitive mode**:* In this mode, the level of trust of MNOs to the Broker is lower than in the cooperative mode, which implies that MNOs do not reveal their costs. Instead, each MNO announces the price charged for each VM and VT to be enabled. The "coopetition" applies because MNOs cooperate on the basis of the common pricing policy and at the same time compete to *maximize their individual profits*. The effectiveness of the pricing policy is validated by means of numerical results in Section V; it is shown that an MNO deviating from this policy will incur a profit loss.

Pricing policy. Motivated by the recent literature (e.g. [16]), we assume that MNOs follow a pay-as-you-go pricing policy which dynamically adapts prices with the availability of resources. In particular, the price ($\$/sec$) that MNO i charges for each VM ν hosted in its domain increases with the amount of allocated resources c_ν, as well as with the amount of its residual resources $\tilde{C}_i$ at the given time. The less the available resources, the higher the price per unit:

$$p_i(c_\nu) = c_\nu\ p_{i,co} \left[1 + \frac{\log(C_i - \tilde{C}_i + c_\nu)}{\log(C_i)} \right], \tag{15}$$

where $p_{i,co}$ denotes the minimum price that an MNO i charges for a unit of computational resources, while the logarithmic part represents the additional price that will be charged per requested computational unit, considering the resource availability and the amount of requested resources. Similarly, the price that MNO i charges for each VT τ_r over edge e_{ji} is given by a function $p_{ji}(b_{\tau_r})$ that considers bandwidth resources.

Local embedding sub-problems. Based on the resource availability and prices announced, the Broker solves multiple *local* embedding sub-problems, one per MNO, following a round-robin approach. The objective of MNO i's sub-problem is the maximization of its individual profit when considering only the requests $\mathcal{R}_i \subseteq \mathcal{R}$ that come from its own customers. The *local* individual profit $LU_i(\mathbf{X}_i, \mathbf{Y}_i)$ of MNO i captures (i) the payments that i receives by the Application Providers for serving requests in $\mathcal{R}_i$, (ii) the cost of i for serving part of these requests and (iii) the compensations that i has to pay to other MNOs for contributing resources to the other parts.

$$\begin{aligned} LU_i(\mathbf{X}_i, \mathbf{Y}_i) = \sum_{r \in \mathcal{R}_i} \Bigg[& \sum_{\pi \in \mathcal{P}_i} y_{\tau_r,\pi}\ \hat{p}_r - \\ & - \sum_{\nu \in \mathcal{V}_r} \Big(x_{\nu,i}\ \kappa_i(c_\nu) + \sum_{j \in \mathcal{I} \setminus \{i\}} x_{\nu,j}\ p_j(c_\nu) \Big) \\ & - \sum_{\pi \in \mathcal{P}_i} y_{\tau_r,\pi} \Big(z(e_{ii}, \pi)\ \kappa_{ii}(b_{\tau_r}) + \sum_{e_{jj'} \in \pi \setminus \{e_{ii}\}} p_{jj'}(b_{\tau_r}) \Big) \Bigg], \end{aligned} \tag{16}$$

The sub-problem of each MNO i is formulated as a MIP that aims to maximize $LU_i(\mathbf{X}_i, \mathbf{Y}_i)$, subject to the same constraints as in cooperative mode, but only for the requests in $\mathcal{R}_i$ and paths $\mathcal{P}_i \subseteq \mathcal{P}$ that include i as a source:

$$\max_{\mathbf{X}_i, \mathbf{Y}_i} LU_i(\mathbf{X}_i, \mathbf{Y}_i),\ \text{s.t. } (6) - (11). \tag{17}$$

The *global* individual profit of an MNO i is estimated by adding to $LU_i(\mathbf{X}_i, \mathbf{Y}_i)$ the compensations that i receives from other MNOs for serving part of their requests.

B. Peer-to-Peer Approach

This approach is suitable when the MNOs do not trust another entity (e.g. a Broker) to perform the embedding for them. Thus, MNOs perform the multi-MNO embedding themselves, through bilateral interactions. Each MNO i determines the embedding for the requests coming from its own customers, $\mathcal{R}_i$, based on information MNOs exchange in terms of their service capabilities and resource pricing. Next, we introduce our information sharing mechanism and discuss how the coopetitive mode is applied in the peer-to-peer approach.

*1) **Information sharing mechanism**:* Inspired by Border Gateway Protocol, we assume each MNO exchanges information with its neighboring MNOs in the form of physical network paths that can support a certain level of QoS and reach a specific geographic region. Each path is a sub-graph of G and it is characterized by an: (i) estimated end-to-end latency, (ii) estimated throughput and (iii) average price *per unit* of computational and network resources. A path is *feasible* for hosting a network slice of certain type and quality class, if the values of estimated end-to-end latency and throughput satisfy the requirements of this request. When multiple feasible paths are available for, then the *preferable* path is the one with the lowest average price per unit of resources. Each MNO i builds and maintains a *table of preferable paths* for all potential combinations of destination geographic region L^{dst}, type t and quality class q. If we use α to denote the number of alternative paths maintained per combination, the total number of paths maintained per MNO is given by $\alpha|\mathcal{L}-1||\mathcal{T}||\mathcal{Q}|$.

Next, we present the constituent elements of our information sharing mechanism.

Path augmentation. Assuming that MNO i receives a path π' from a neighboring MNO j, over which a network slice

of type t and quality q can be embedded for reaching region L^{dst}, MNO i then performs a path augmentation for including its own domain, i.e. it generates path $\pi = \pi^{'} \cup \{e_{ji}, e_{ii}\}$, and estimates the characteristics of the augmented path:

(i) *Estimated end-to-end latency.* The latency of the augmented path π is estimated by adding up the network latency in MNO i's domain, without considering any processing delay which is only estimated once by the MNO who initiated the path creation and it is already captured in $\pi^{'}$. Path π is feasible for placing a network slice of type t and quality q, if the estimated latency does not exceed $D(t, q)$.

(ii) *Estimated throughput.* For estimating the throughput of path π, MNO i should identify the bottleneck in its domain and compare it with the estimated throughput of path $\pi^{'}$. This is done by selecting the minimum of: (a) estimated throughput of path $\pi^{'}$, (b) residual bandwidth of links $\{e_{ji}, e_{ii}\}$ and (c) throughput per VM, if all VMs of a network slice of type t and quality q are deployed in i's domain (worst case scenario) and the resources are evenly distributed among them. Path π is feasible for embedding a network slice of type t and quality q if the estimated throughput exceeds $B(t, q)$.

(iii) *Average price.* Having knowledge about the average price of computational and network resource units across path $\pi^{'}$, MNO i estimates average prices across path π through a weighted average, based on the number of nodes and edges in $\pi^{'}$, also including its own unit prices.

Cheapest path update and selection rule. A entry of preferable paths table is updated once an MNO receives a path that indicates a new *preferable* path. Let $\mathbf{\Pi}$ denote the set of all augmented paths that MNO i has generated after receiving updates from its neighbors, which are *feasible* for deploying network slices of type t and quality class q that reach region L^{dst}. The *cheapest* path π^* is selected by estimating the aggregate price of the resources required when deploying a network slice of type t and quality q over each path $\pi \in \mathbf{\Pi}$. The cheapest path selection rule is the tool that incentivizes the truthfulness of MNOs when announcing prices to their neighbors because it leads MNOs to be competitive in terms of prices in order to have a higher chance of being selected.

Path forwarding. After updating an entry on its table, an MNO *pushes* the new preferable path to its neighbors.

*2) **Peer-to-peer Coopetitive mode**:* Each MNO solves its local embedding sub-problem (17), by utilizing *only* the paths that are available in its table of preferable paths. The MNOs solve their sub-problems sequentially, e.g., in a round-robin fashion. Note that some requests may not be served in the first round, due to lack of feasible paths. However, after the path updates that happen after solving sub-problems, feasible paths may be revealed and utilized in the next rounds.

V. Numerical Evaluation

We develop a Python program that simulates an environment of multiple interconnected MNOs, jointly offering network slices to Application Providers. We generate multiple random topologies of MNOs dispersed in different regions, with randomly generated requests arriving at each MNO for different network slice types and qualities. Our numerical study focuses on assessment of the performance of our policies in terms of *total* and *individual* MNO profits. We have studied multiple simulations setups with variable configurations and the extracted results are the averages over 100 runs per setup.

A. Simulation Setup

Topology, resource capacity and cost. We generate random topologies of $[5, 10, 15]$ MNOs, dispersed in $[3, 5, 6, 10]$ geographic regions. We assume that each of the MNOs maintains 100 CPU cores, while the bandwidth of each network link is 70 $Gbps$. In addition, each network link is characterized by a random latency from 1 to 3 $msec$. The cost per CPU core is set to 0.3 $\$/core/hour$, while the cost per unit of network resource is set to 0.04 $\$/Gbps/hour$. We utilized these unit cost values for extracting the minimum price for a unit of resource by multiplying them with a factor of 3.

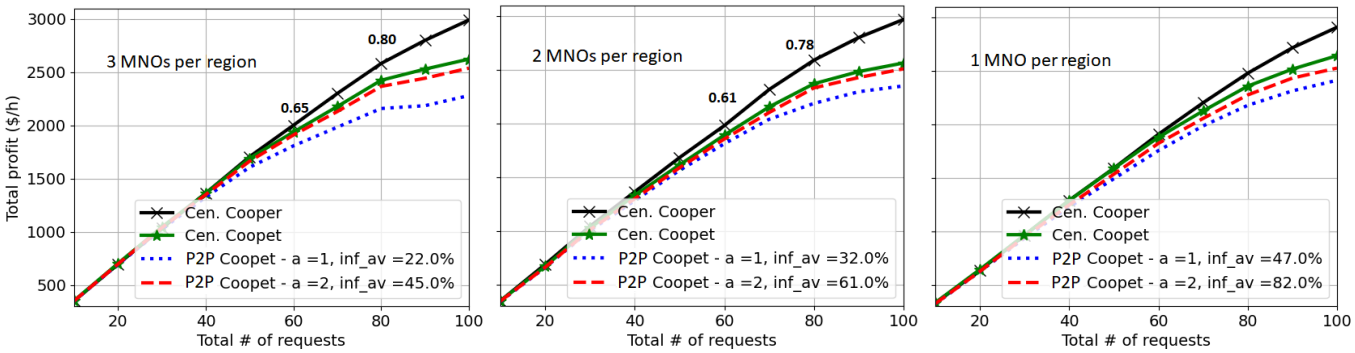
Services and QoS target values. We assume that two different network slice types are offered, i.e. uRLLC and eMBB, in two quality classes, namely the standard and premium one. Each service type consists of a random number of chained VNFs, from a total of 20 such VNFs that are available in each MNO. The throughput and latency target values for a uRLLC slice are set to 5 $Gbps$ and 10 $msec$ for the standard quality class and to 10 $Gbps$ and 5 $msec$ for the premium class. The throughput and latency target values for an eMBB slice are set to 10 $Gbps$ and 20 $msec$ for the standard quality class and to 20 $Gbps$ and 15 $msec$ for the premium class.

Services requests. For each of the generated topologies, we create $[10, 20, ..., 100]$ requests. For each request we randomly select the network slice type and quality class, the MNO that receives the request, the source and destination geographic regions, etc. The customers' willingness-to-pay is a random value between 30 and 40 $\$/request/hour$.

B. Numerical Results

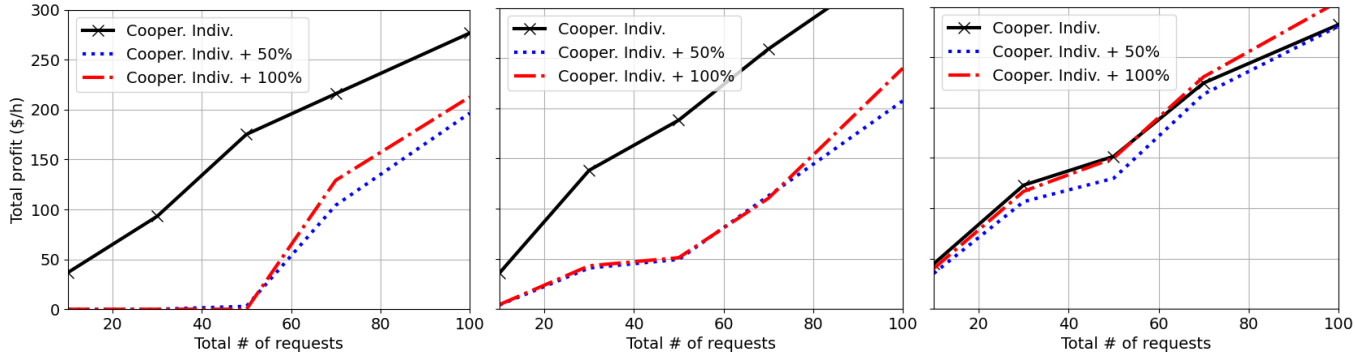
*1) **Total Profit**:* The total profit of MNOs is higher under the centralized cooperative mode, since the Broker solves a total profit maximization problem having full knowledge about the potential paths, resource availability and costs. However, in Fig. 4, we can observe that the centralized coopetitive mode achieves a total profit that is quite close to the cooperative one when infrastructure utilization is $<\sim 0.65$. As the infrastructure utilization increases, so does the performance difference between these two modes. This difference is only up to 5% for a utilization value that does not exceed ~ 0.8 (Fig. 4), but it reaches $\sim 15\%$ as the average utilization approaches 1.

Number of MNOs per geographic region. When the number of MNOs per region is greater than 1, the centralized cooperative mode further improves its performance against the centralized coopetitive one. In Fig. 4a, we observe that when 3 MNOs have presence in each region, the maximum difference reaches 15%. On the other hand, when 1 MNO is present in each region, the maximum difference is 9%. This happens because in coopetitive mode the Broker investigates only the paths that include as source the MNO that received



(a) 10 MNOs, 3 Geographic regions (b) 10 MNOs, 5 Geographic regions (c) 10 MNOs, 10 Geographic regions

Fig. 4. Total profit of MNOs under all proposed approaches and modes, for multiple random topologies of 10 MNOs and requests ranging for 10 to 100. Evaluation of the peer-to-peer approach for different α's, i.e. number of paths maintained per tuple of destination region, service type and quality class.



(a) 10 MNOs, 3 Geographic regions (b) 10 MNOs, 5 Geographic regions (c) 10 MNOs, 10 Geographic regions

Fig. 5. Individual profit of an MNO that acts strategically under the cooperative mode, i.e. by declaring higher cost than its actual one.

the request, while in cooperative mode all potential alternatives are investigated. As the number of MNOs per region increases the cooperative mode exhibits greater optimization potential.

Information availability impact. The peer-to-peer coopetitive mode can achieve a total profit equal (or close to) the centralized coopetitive one, which is its upper bound. In Fig. 4, we show that the performance of the peer-to-peer coopetitive mode depends on the size of the table of "preferable" paths, which is determined by parameter α, and the number of geographic regions. In Fig 4a, we can observe that by selecting a value of α that achieves information availability (in terms of potential paths) greater than 45% of full knowledge, the peer-to-peer coopetitive mode performs really close to the upper bound. As the number of MNOs per region increases, a higher value for parameter α is needed for achieving the same percentage of information availability. For instance, when $\alpha = 2$, we achieve 45%, 61% and 82% information availability for 3, 2 and 1 MNOs per geographic region, respectively.

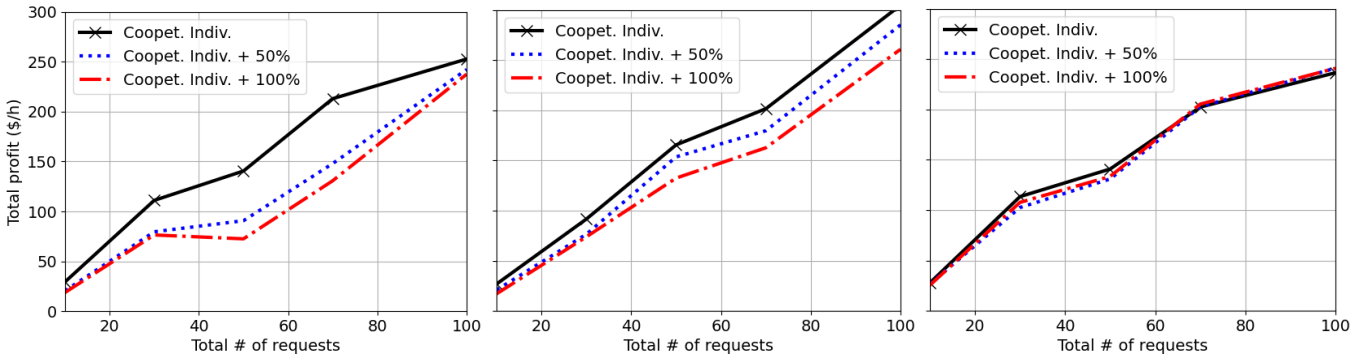
*2) **Individual Profits**:* Then, we show that if an MNO complies with the policies defined under each approach and mode, its individual profit is maximized.

Impact of untruthfulness. Focusing on the cooperative mode, we examine whether an MNO has the incentive to declare higher cost than its actual one, aiming to increase its profits, when all others are truthful. Figure 5 shows the individual profit of a truthful MNO and when it announces a cost that is 1.5 and 2 times greater than its actual one. In Fig. 5a and Fig. 5b, we observe that when more than 1 MNOs are available in each geographic region, the untruthful MNO will always have profit loss. The only case where an untruthful MNO can generate higher profit is when the Broker does not have the option of alternative MNOs for provisioning a service. As we observe in Fig. 5c, this applies when only 1 MNO is present in each region and for high values of infrastructure utilization. MNO cannot have knowledge about others' utilization, thus an untruthful behavior cannot be selected.

Impact of strategic pricing. Figure 6 shows the individual profit of an MNO that acts strategically in coopetitive mode, i.e. unilaterally deviates from the common pricing formula and charges a higher price. When multiple MNOs are present in each geographic region, the MNO that acts strategically will always have profit loss, because MNOs with lower price will be selected instead. On the other hand, when only one MNO is present in each region, the strategic MNO can generate slightly higher profit. However, such a behavior may lead to the provisioning of less requests in total, due to a higher service prices than the ones customers can afford. Though, this strategic behavior is also prevented since an MNO cannot have full knowledge about others' infrastructure utilization.

C. Lessons Learned from the Numerical Analysis

(i) When the demand and resource utilization are *low*, the peer-to-peer coopetitive mode performs really close to the centralized schemes. In such cases, the MNOs will choose to adopt the peer-to-peer coopetitive mode since the risk of



(a) 10 MNOs, 3 Geographic regions (b) 10 MNOs, 5 Geographic regions (c) 10 MNOs, 10 Geographic regions

Fig. 6. Individual profit of an MNO that strategically announces higher prices than the ones indicated by the common formula.

resource overloading and potential loss of profit is low. When the demand and resource utilization are *high*, the MNOs have a strong incentive to put their trust concerns aside and cooperate in order to generate significantly more profits. Given that 5G networks are not expected to be highly utilized in order to achieve the extreme QoS required the peer-to-peer coopetitive mode is expected to be most broadly adopted.

(ii) When the number of MNOs per geographic region is limited to 1, the performance advantage of centralized cooperative mode from the coopetitive ones diminishes. This implies that when only a monopolistic MNO is active in each region, the optimization potential of the cooperative mode is limited thus the coopetitive mode will be preferable.

(iii) Under the peer-to-peer coopetitive mode, the high degree of information availability among MNOs results to a performance that the Broker can achieve under the coopetitive mode. Not sharing information may not significantly reduce the MNOs' profits when the demand is low. However, it is not beneficial when the demand is high, since the need for an efficient resource allocation across all MNOs is also high.

(iv) In the majority of cases, cooperative mode incentivizes MNOs' truthfulness when declaring costs, while our coopetitive mode prevents the strategic pricing.

VI. Conclusions

In this paper, we present economic models and policies for the embedding of multi-MNO network slices under centralized and peer-to-peer approaches. We formulate this problem as a MIP, which can be solved under both cooperative and coopetitive modes, either by a Broker (centralized) or by an MNO (peer-to-peer). Our polices guarantee that the profit of MNOs is maximized when they comply with the policies and rules defined under the approaches and modes considered. The results show that sophisticated approaches such as our peer-to-peer coopetitive mode can achieve a total profit that is very close to that of centralized cooperative (benchmark). Simplistic distributed approaches have inferior performance. Moreover, when applying our policies, untruthful or strategic behavior of MNOs is averted. Directions for future work include the extension of our model to support the provisioning of a network slice over multiple paths and the evaluation of our peer-to-peer approach under inaccurate information.

Acknowledgment

This work is part of the EU H2020 5G PPP Phase 3 5G-VINNI project that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 815279. I. Koutsopoulos acknowledges support from the CHIST-ERA grant CHIST-ERA- 18-SDCDN-004 (grant number T11EPA4-00056) through the General Secretariat for Research and Innovation (GSRI).

References

[1] M. Chowdhury, M. R. Rahman, and R. Boutaba, "Vineyard: Virtual network embedding algorithms with coordinated node and link mapping," *IEEE Trans. on Networking*, vol. 20, no. 1, pp. 206–219, 2012.

[2] M. C. Luizelli et al., "Piecing together the nfv provisioning puzzle: Efficient placement and chaining of virtual network functions," in *Proc. of IFIP IM*, 2015.

[3] M. Leconte et. al, "A resource allocation framework for network slicing," in *Proc. of IEEE INFOCOM*, 2018.

[4] S. D'Oro, L. Galluccio, S. Palazzo, and G. Schembra, "A game theoretic approach for distributed resource allocation and orchestration of softwarized networks," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 3, pp. 721–735, 2017.

[5] ——, "Exploiting congestion games to achieve distributed service chaining in nfv networks," *IEEE Journal on selected areas in communications*, vol. 35, no. 2, pp. 407–420, 2017.

[6] P. Caballero, A. Banchs, G. De Veciana, and X. Costa-Pérez, "Network slicing games: Enabling customization in multi-tenant mobile networks," *IEEE Trans. on Networking*, vol. 27, no. 2, pp. 662–675, 2019.

[7] R. Guerzoni et al., "A novel approach to virtual networks embedding for sdn management and orchestration," in *Proc. of IEEE NOMS*, 2014.

[8] M. Chowdhury, F. Samuel, and R. Boutaba, "Polyvine: policy-based virtual network embedding across multiple domains," in *Proc. of ACM SIGCOMM*, 2010.

[9] I. Houidi, W. Louati, W. B. Ameur, and D. Zeghlache, "Virtual network provisioning across multiple substrate networks," *Computer Networks*, vol. 55, no. 4, pp. 1011–1023, 2011.

[10] I. Vaishnavi, R. Guerzoni, and R. Trivisonno, "Recursive, hierarchical embedding of virtual infrastructure in multi-domain substrates," in *Proc of IEEE NetSoft*, 2015.

[11] T. Mano et al., "Efficient virtual network optimization across multiple domains without revealing private information," *IEEE Trans. on Network and Service Management*, vol. 13, no. 3, pp. 477–488, 2016.

[12] M. F. Bari, S. R. Chowdhury, R. Ahmed, and R. Boutaba, "On orchestrating virtual network functions," in *Proc. of CNSM*, 2015.

[13] G. S. Paschos, M. A. Abdullah, and S. Vassilaras, "Network slicing with splittable flows is hard," in *Proc. of IEEE PIMRC*, 2018.

[14] R. Trivisonno et. al, "Network resource management and qos in sdn-enabled 5g systems," in *Proc. of IEEE GLOBECOM*, 2015.

[15] L. He and J. Walrand, "Pricing and revenue sharing strategies for internet service providers," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 5, pp. 942–951, 2006.

[16] L. Toka, J. Tapolcai, G. Darzanos, and B. Sonkoly, "On pricing of 5g services," in *Proc. of GLOBECOM*, 2017.