Orchestrating End-to-end Slices in 5G Networks

Davit Harutyunyan*, Riccardo Fedrizzi*, Nashid Shahriar§, Raouf Boutaba§ and Roberto Riggio*
*Wireless and Networked Systems, FBK, Italy; Email: d.harutyunyan,rfedrizzi,rriggio@fbk.eu
§David R. Cheriton School of Computer Science, University of Waterloo, Canada; Email: nshahria,rboutaba@uwaterloo.ca

***Abstract*—5G networks are characterized by massive device connectivity, supporting a wide range of novel applications with their diverse Quality of Service (QoS) requirements. This poses a challenge since 5G as *one-fits-all* technology has to simultaneously address all these requirements. Network slicing has been proposed to cope with this challenge, calling for efficient slicing and slice placement strategies in order to ensure that the slice requirements (e.g., latency, data rate) are met, while the network resources are utilized in the most optimal manner.**

In this paper, we compare different end-to-end (E2E) slice placement strategies by formulating and solving a Mixed Integer Linear Programming (MILP) slice placement problem and study their trade-offs. E2E slice requests are modelled as Service Functions Chains (SFC), in which each core network and radio access network component is represented as a Virtual Network Function (VNF). Based on the analysis of the results, we then propose a slice placement heuristic algorithm whose objective is to minimize the number of VNF migrations in the network and their impact onto the slices while, at the same time, optimizing the network utilization and making sure that the QoS requirements of the considered slice requests are satisfied. The results of the simulations demonstrate the efficiency of the proposed algorithm.

***Index Terms*—5G, End-to-end Network Slicing, Mobile Edge Computing, Service Function Chain Placement.**

I. Introduction

The fifth generations mobile network technology (5G) promises to support high connection density, unprecedented data rates, and ultra-low latencies [1]. This lays the ground for many novel applications such as virtual/augmented reality, real-time interactive online gaming, autonomous driving and so forth, whose requirements cover various permutations of the aforementioned parameters, which have been impossible with the previous generations of mobile network technologies. The requirements of such applications are usually classified into three main network service categories: Enhanced Mobile Broadband (eMBB), Massive Machine-type Communication (mMTC) and Ultra-reliable Low Latency Communications (URLLC) [2]. Since these network service categories have versatile requirements and prioritize their key performance indicators in different ways, one of the challenges posed on 5G mobile network technology is to simultaneously address all the requirements of various applications that may belong to different network service categories [3].

This challenge is tackled owing to Network Function Virtualization (NFV) technology, which is expected to play a key role in the realization of the 5G network [4], [5]. Indeed, recent advances in NFV enables Mobile Network Operators (MNO), also called Infrastructure Providers (InP), to share their Radio Access Network (RAN) resources (e.g., RF bands, RF frontends) as well as the core network among, for example, multiple Mobile Virtual Network Operators (MVNO[1]). Better yet, NFV enables MNOs to allocate dedicated RAN and core network resources, dubbed as network slicing, guaranteeing resource isolation between different MVNOs, which is one of the challenges of network slicing [6]. Thanks to the NFV technology, both the RAN components (i.e., gNBs, which are in charge of performing users' baseband signal processing) and the 5G core network components (e.g., the ones performing Access and Mobility Management Functions (AMF), Session Management Functions (SMF), User Plane Functions (UPF) [7]) in a network slice can be represented as Virtualized Network Functions (VNF) chained together in a particular order forming an end-to-end (E2E) slice/Service Function Chain (SFC). This SFC complies the requirements of 3GPP for both User Plane (UP) traffic flow, which uses UPF, and Control Plane (CP) traffic, which flows through AMF, SMF and UPF[2] [8]. Each E2E slice/SFC request (e.g., eMBB, mMTC, URLLC) can be characterized by the required number of gNBs, traffic per gNB and one-way CP/UP latency.

Multi-access Edge Computing (MEC) is yet another technology expected to be widely adopted in 5G networks [9], [10]. The main idea behind MEC technology is to bring the content and the computational power closer to the end-users (co-locate with the gNBs, in the best case), making up a light Data Center (DC), curtailing the round-trip service provisioning latency and alleviating the transport network utilization. In this scenario, all the physical nodes in the mobile network can be thought of DCs, whose capacity depends on their type (e.g., core nodes, gNBs, aggregation points for gNBs such as a centralized unit in the Cloud-RAN scenario [11]). The closer is the DC to the core network, the more is its capacity and, consequently, the more SFCs it can host.

In the aforementioned mobile network deployment scenario, the job of an InP receiving E2E slice requests would be to embed the requests onto the substrate network such as to (i) make sure that the slice requirements such as data rate and latency are satisfied, (ii) guarantee that the slices are isolated and (iii) seek to utilize the substrate resources in the most efficient manner. All these considerations make E2E slice embedding a non-trivial task also because of non-uniform availability of computing (vCPU), memory (vRAM) and storage (vSTO) resources at the Data Centers (DC) located at different layers of the MNO's network. Thus, the optimal

Research leading to these results received funding from the European Unions H2020 Research and Innovation Action under Grant Agreement H2020–ICT–761592 (5G–ESSENCE Project).

[1]MVNOs are wireless communication service providers that own no mobile network infrastructures and, therefore, exploit other MNOs' infrastructures to provide their communication services.

[2]While a number of 5G core network components exist [7], we consider only AMF, SMF and UPF as main components for the UP/CP traffic flow.

978-3-903176-24-9

E2E slice embedding requires meticulous consideration of a number of factors [12].

While there is a significant amount of works published on the core network component placement problem in 4G networks [13], [14], there are a few studies considering the same problem in 5G networks [15], [16]. On the other hand, there are several papers proposing RAN slicing/sharing strategies [17], [18]. In the 5G networks, the research to date has tended to focus either on the RAN slicing or the core network slicing and placement problem. As opposed to these works, in this paper, we consider 5G E2E network slicing, formulating its placement problem, which is modelled as a Virtual Network Embedding (VNE) problem and is solved using Mixed Integer Linear Programming (MILP) techniques. Specifically, we draw a comparison between E2E slice placement strategies that aim at minimizing, respectively, the E2E slice embedding cost, the link bandwidth consumption and the number of VNF migrations. Additionally, in order to tackle the scalability issue of the MILP-based algorithms, we propose a heuristic algorithm that follows the last embedding approach, which demonstrates to be the most efficient in utilizing network resources, resulting in the minimum number of VNF migrations.

The rest of this paper is structured as follows. The related work is discussed in Sec. II. The problem statement along with the mobile network and E2E slice request models are introduced in Sec. III. The MILP problem formulation and the heuristic are presented in Sec. IV. The numerical results are reported in Sec. V. Finally, Sec. VI draws the conclusions.

II. Related work

RAN Slicing. RAN slicing allows multiple tenants (e.g., MVNOs) to share the same physical resources (spectrum and/or compute) through different type of isolation, scheduling, and virtualization techniques [19], [20]. To achieve RAN slicing, [21] uses virtualization through hypervisors that can host virtual eNodeBs of different MVNOs and allocate Physical Resource Blocks (PRBs) to virtual nodes based on predefined contracts. Among the other approaches, [22] proposes a network-wide Radio Resource Management (RRM) framework for RAN sharing, whereas [23] advocates for application-oriented RAN sharing. In contrast, [24] introduces a two-level scheduler to share PRBs among slices. In a multi-cell network, [25] analyzes four RAN slicing approaches that differ in the RRM functions used to split RAN resources among slices. By modeling spectrum as two-dimensional (frequency and time) grid, [26] uses a Karnaugh-map based algorithm to embed slices on wireless resources. A PRB allocation strategy is proposed in [18] to maximize the transmission rate achieved by each user. Recently, [17] presents a RAN slicing approach that allocates dedicated chunks of PRBs to small cells requested by MVNOs leveraging an optimal RAN functional split.

Core Network Component Placement in 4G/LTE. A sizable body of work has been published on virtual Evolved Packet Core (vEPC) placement in 4G/LTE where EPC components such as Serving/Packet Gateway (S/P-GW), Home Subscriber Server (HSS) and Mobility Management Entity (MME) are represented as VNFs [13], [14], [27]–[31]. Among them, SoftEPC [13] enables on–demand and load–aware instantiation of EPC functions at appropriate locations to place the frequently used functions close to users. In [29], an MILP formulation is proposed to embed SFCs composed of VNFs of EPC minimizing the cost of node and link resources, while satisfying the latency constraints. The authors in [14] formulate the problem of vEPC mapping in a federated cloud using a coalition formation game. The same authors study a service–aware PGW placement problem to reduce the network operators' cost [30]. KLEIN [31] addresses the problem of managing vEPC resources to distribute load across vEPC instances in different DCs. To improve EPC's scalability, [32] and [33] propose to decouple SGW and PGW into control plane (SGW-C and PGW-C) and user plane (SGW-U and PGW-U) components. The benefit of such decoupling is that control plane functions can be offloaded to DCs as VNFs and SDN can be used to route traffic through VNFs.

Core Network Component Placement in 5G. To leverage the full benefits of NFV, a new service-based architecture is proposed for the 5G mobile core network [20], [34]. Based on this architecture, [16] presents an optimization model for the placement of statefull VNFs to simultaneously minimize the state transfer frequency and network latency. In contrast, [15], [35] abstract a 5G core slice as a set of SFCs and addresses cost-optimal deployment of slices on cross-domain DCs.

E2E Slicing in 5G. To date, several studies including [36]–[38] have developed prototypes for E2E slicing in 5G mobile networks. In terms of resource orchestration, [39] proposes a fully distributed resource allocation scheme to realize an E2E slice across 3-tier DCs. However, [39] assumes the placement of VNFs of an E2E slice on DCs is pre-determined and the resource allocation is done using auction theory. In contrast, we consider full-fledged resource orchestration of 5G E2E slices that consist of both RAN and core components.

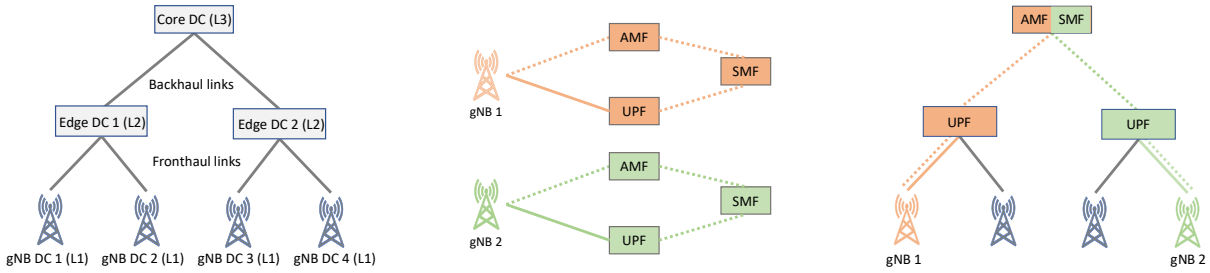
III. Network Model

This section formally states the problem and details the substrate network model along with the slice request model.

A. Problem Statement

Figure 1a depicts the reference network architecture for the E2E slice placement problem in 5G mobile networks. Each slice request is represented as an SFC that encompasses both the RAN (i.e., gNBs) as well as the core network, whose main components (i.e., AFM, UPF and SMF) are represented as VNFs. Two types of nodes are distinguished in the mobile network: gNBs and non–gNBs with the latter represented as rectangles. While all the nodes can host core network component VNFs, only gNBs can accommodate virtual gNBs in the slice requests. Thus, all the nodes can be considered as DCs some of which have baseband processing capabilities (i.e., gNBs), like in [40]. DCs are located either in Layer 1 (L1), Layer 2 (L2), or in Layer 3 (L3). The lower is the DC location, the less is its resources (e.g., vCPU, vRAM and (vSTO)), which, in turn, means that the fewer VNFs can be hosted by that DC. This is justified by the mushrooming number of small cell gNBs that are expected to be deployed in order to meet the expected traffic demand [41].

Figure 1b illustrates a sample E2E slice/SFC request, which can be made by MVNOs. The considered SFC is composed of two gNBs and their corresponding core network component



(a) Mobile network topology. (b) Sample eMBB slice request. (c) Slice placement.

Fig. 1: Sample mobile network, slice request and slice placement. While all nodes in Fig. 1a are DCs, only some of them are also gNBs. Green, orange solid and dotted lines in Fig. 1b and Fig. 1c represent, respectively, UP and CP traffic flows.

TABLE I: Mobile network parameters

Parameter	Description
G_{dc}	Mobile network graph.
N_{dc}	All nodes/DCs in G_{dc}.
N_{gnb}	Set of gNBs in G_{dc}.
$N_{ist}^{d,v}$	Set of instances of VNF $v \in V$ on DC $d \in N_{dc}$.
V	Set of core network VNFs (i.e., AMF, UPF, SMF).
Γ	Set of resource types (i.e., vCPU, vRAM, vSTO).
E_{dc}	Set of substrate links in G_{dc}.
$\omega_{bwt}(e^{nm})$	Capacity of the link $e^{nm} \in E_{dc}$.
$\omega_{tr}(g)$	Maximum traffic supported by the gNB $g \in N_{gnb}$.
$\omega_{\gamma}^{v}(d,i)$	$\gamma \in \Gamma$ resource of i^{th} instance of VNF $v \in V$ on DC d.
$\omega_{gnb}^{v}(d,i)$	# of gNBs that i^{th} instance of VNF v supports on DC d.
$\omega_{\gamma}(d)$	$\gamma \in \Gamma$ resource of DC $d \in N_{dc}$.
$loc(d)$	Geographical location of the DC $d \in N_{dc}$.
$\delta(g)$	Coverage radius of the gNB $g \in N_{gnb}$ (in meters).
μ_s, μ_b	Small and big positive numbers, respectively.

VNFs (e.g., AMF, UPF and SMF). Depending on its demand, each gNB in the request generates a certain amount of CP and UP traffic, which are represented by dotted and solid lines, respectively. As can be observed, the CP traffic follows the path gNB $\rightarrow$ AMF $\rightarrow$ SMF $\rightarrow$ UPF; whereas, the UP traffic is directly transmitted to the host UPF i.e. gNB $\rightarrow$ UPF [8]. Apart from the CP and UP traffic demand, each gNB in the slice request is also characterized by its CP and UP latency requirement for their aforementioned one-way paths, respectively. It is worthwhile to note that each gNB in the slice request has to be connected to a single core network component of its type (e.g., AMF, UPF and SMF). Receiving this kind of SFC requests, the InP shall embed the request to the substrate network and allocate sufficient amount of node and link resources, while making sure that the requirements of the SFCs are satisfied and the network resources are used in an efficient manner. Depending on the requested SFC requirements and the utilization of network resources, different mapping strategies may be possible each minimizing a certain cost function. The problem of E2E slice placement can be formally stated as follows:

Given: a 5G mobile network composed of hierarchical DCs, each having a certain amount of resources (i.e., vCPU, vRAM, vSTO), some of which are collocated with gNBs, each supporting a certain amount of traffic, the transport network topology with the capacity of each link. Additionally, given SFC requests each having a certain CP and UP latency requirements to be imposed on each of its gNB, which also has a certain traffic demand.

Find: SFC placements, that is, the placement of both gNBs and their core network VNFs, and the resource allocation in the substrate network.

Objectives: minimize (i) the bandwidth consumption in the transport network, (ii) the service provisioning cost, and (iii) the number of VNF migrations and their effect onto the slices.

B. Mobile Network Model

Let $G_{dc} = (N_{dc}, E_{dc})$ be an *undirected* graph modelling the substrate mobile network, where N_{dc} is the set of DCs. A subset of DCs $N_{gnb} \subseteq N_{dc}$ is substrate gNBs. E_{dc} is the set of substrate links, which can be either fronthaul links (FH), interconnecting L1 DCs with L2 DCs, or backhaul links (BH), interconnecting L2 DCs with L3 DCs. An edge $e^{nm} \in E_{dc}$ exists if and only if a connection exists between $n, m \in N_{dc}$. Each gNB $g \in N_{gnb}$ is characterized by its maximum supportable traffic $\omega_{tr}(g)$, which is computed based on its total bandwidth (i.e., the available PRBs) and Multiple-input and Multiple-output (MIMO) antenna configurations while, assuming average modulation order of 16QAM and 25% of overhead of various channels, reference and synchronization signals, and channel coding, like in [42]. Each DC $d \in N_{dc}$ and instance $i \in N_{ist}^{d,v}$ of the VNF $v \in V$ instantiated on the DC d is characterized by its available resources $\omega_{\gamma}(d)$ and $\omega_{\gamma}^{v}(d,i)$, respectively, in which $\gamma \in \Gamma$ represents the resource type e.g., vCPU, vRAM, vSTO. For the sake of simplicity, it is assumed that all core network VNFs require the same amount of resources to be spawned/instantiated. However, the model can be easily extended to consider VNF instances with variable resource requirements and capacities. It is worth mentioning that this model also tackles the case where, due to high traffic demand, multiple instances of the same VNF type (e.g., AMF, SMF, UPF) need to be instantiated for the same slice request. Each DC $d \in N_{dc}$ is associated with a geographic location $loc(d)$, as x, y coordinates while each gNB $g \in N_{gnb}$ is also associated with a coverage radius of $\delta(g)$, in meters. Another weight $\omega_{bwt}(e^{nm})$ is assigned to each link $e^{nm} \in E_{dc} : \omega_{bwt}(e^{nm}) \in \mathbb{N}^{+}$ representing the capacity (in Gbps) of the substrate link connecting the DCs n and m. Table I summarizes the mobile network parameters.

C. E2E Slice/SFC Request Model

The E2E slice request is modelled as an *undirected* graph $G_{slc}^{k} = (N_{slc}^{k}, E_{slc}^{k})$ where $N_{slc}^{k} = N_{ran}^{k} \cup N_{core}^{k}$ is the union

TABLE II: Mobile network slice request parameters

Parameter	Description
G^k_{slc}	Mobile network slice graph.
N^k_{slc}	Set of all the nodes in the requested slice.
N^k_{ran}	Set of gNBs in the requested slice.
N^k_{core}	Set of core network component VNFs in requested slice.
E^k_{slc}	Set of all virtual links in the requested slice $k \in K$.
$\underline{E}^k_{slc}, \overline{E}^k_{slc}$	Set of virtual links, respectively, for UP/CP traffic in k.
K	Set of E2E slice requests.
$\omega_{tr}(r)$	Traffic demand of gNB $r \in N^k_{ran}$ of slice $k \in K$.
$\omega^v_\gamma(c)$	γ resource demand of VNF $c \in N^{k,v}_{core}$ of type $v \in V$.
$\omega_{bwt}(e')$	Data rate demand of $e' \in E^k_{slc}$ of slice $k \in K$.
$loc(r)$	Geographical location of gNB $r \in N^k_{ran}$ of slice $k \in K$.

of the set of gNBs and their corresponding core network component VNFs, while E^k_{slc} is the set of virtual links in the E2E slice request $k \in K$. Each E2E slice requested by an MVNO has its type (e.g., URLLC, eMBB or mMTC), which, in turn, has its QoS requirement expressed in terms of maximum acceptable UP/CP latency ($\tau^k_{up,cp}(r)$) and traffic demand ($\omega_{tr}(r)$) per requested gNB r, where the latter can be estimated by considering the average number of users and their traffic demand. UP latency is computed as follows: $\tau^k_{up} = \zeta^{k,e'}_e + \tau^{up}_{exc}$, where $\zeta^{k,e'}_e$ and τ^{up}_{exc} are, respectively, the FH/BH link transmission and propagation time, and the UPF execution time. CP latency is computed as follows: $\tau^k_{cp} = \tau^{prop}_{tx} + \tau^{cp}_{exc}$, where τ^{prop}_{tx} is the propagation time over the FH/BH transmission links, while τ^{cp}_{exc} is the execution time of CP events on AMF and SMF. It is worthwhile to note that no transmission time is considered for the CP traffic since it is negligible compared to the UP traffic [43]. Additionally, each gNB $r \in N^k_{ran}$ in the slice request k, has its desired geographic location $loc(r)$, as x, y coordinates, to be deployed. As for the core network component VNFs N^k_{core}, it is the InP's job to compute the required quantity of each VNF type based on the requested number of gNBs and their traffic demand as well as the available resources on the host DCs. Table II summarizes the slice request parameters.

It is important to mention that while each gNB in the slice request has to be connected to a single VNF of its type (e.g., AMF, UPF, SMF), multiple gNBs of the same slice can be connected to the same VNF instance of their type as long as they have sufficient capacity and their connected number of gNBs is less than the maximum number of supported gNBs. The gNBs belonging to different slice requests, however, cannot be connected to the same VNF instance. This is enforced in order to guarantee isolation between different slices. The actual embedding of the core network VNFs requested by different MVNOs depends on several factors such as the optimization objective, the UP/CP latency requirements of the slice, the availability of substrate network resources, etc. Figure 1c illustrates an example of such an embedding of the eMBB slice request depicted in Figure 1b. The goal of this embedding is to minimize the number of VNF migrations in the network. This objective considers both FH/BH bandwidth consumption cost as well as the VNF instantiation cost, which, in this case, depends on the host DC and the slice type. Such an objective function results in separate UPF instances being placed on L2 DCs, close to their respective gNBs, where the instantiation cost of separate VNFs are justified due to significantly high UP traffic demand in comparison with the CP one. Note that the resources of L1 DCs are saved for serving the forthcoming slice requests that may have stricter UP/CP latency requirements (e.g., URLLC slices). As for the CP VNFs (i.e., AMF and SMF), the gNBs in the slice request share the same AMF and SMF instances that are placed on the same L3 DC since having separate AMF and SMF instances closer to the gNBs would result in a higher VNF instantiation cost, which would not be justified due to negligible CP traffic.

IV. Problem Formulation

Upon receiving SFC requests from MVNOs, the InP shall decide if to accept or to reject the request. In the case of accepting the SFC request, the InP shall make a decision on how to embed the request onto the substrate network so as to satisfy the traffic as well as the UP/CP latency requirements of the slice, while, at the same time, making sure that the substrate network resources are used in an efficient manner. This embedding problem is modeled as a VNE problem, which is *NP*–hard and has been studied extensively in the literature [44], [45]. The embedding process consists of two parts: the node embedding and the link embedding. In the node embedding, each virtual node in the request is mapped to a substrate node (i.e., gNBs and non-gNBs in the substrate network). In the link embedding instead, each virtual link is mapped to a single substrate path. In both cases, the constraints of the nodes and links must be satisfied.

A. MILP Formulation

Before formulating the MILP-based VNE problem, we first need to find the candidate substrate gNBs for each gNB in the slice request. Considering the location $loc(r)$ of the gNB $r \in N^k_{ran}$ along with the location $loc(g)$ and the coverage radius $\delta(g)$ of the substrate gNBs $\forall g \in N_{gnb}$, a cluster of candidate gNBs $\Omega(r)$ for the gNB r can be defined as follows:

$$\Omega(r) = \Big\{ g \in N_{gnb} | dis(loc(g), loc(r)) \leq \delta(g) \Big\} \tag{1}$$

Since the same FH/BH links and VNFs may be used by multiple gNBs, we then define L as mandatory argument to be used in the objective functions to guarantee the accurate estimation of the transmission time over the FH/BH links. It has a very small positive value (μ_s) in order to make sure that the main arguments of the objective functions are not affected. All the variables used in this formulation are summarized in Table III.

$$L = \sum_{k \in K} \sum_{e' \in E^k_{slc}} \sum_{e \in E_{dc}} \mu_s \zeta^{k,e'}_e \tag{2}$$

Using formula (3), we define variable $\Phi^{k,c,c^\star}_{d,v,i} = \Phi^{k,c}_{d,v,i} \Phi^{k,c^\star}_{d,v,i}$, which indicates if the VNFs c and $c^\star$ of type v requested by the slice k are using the same i^{th} instance of the VNF v of DC d assigned to the slice k.

$$\Phi^{k,c}_{d,v,i} + \Phi^{k,c^\star}_{d,v,i} - \Phi^{k,c,c^\star}_{d,v,i} \leq 1 \tag{3}$$

$$\forall k \in K,\ v \in V,\ c \in N^{k,v}_{core}, c^\star \in N^{k,v}_{core},\ d \in N_{dc},\ i \in N^{d,v}_{ist}$$

TABLE III: Binary (Φ) and continuous (ζ) decision variables.

Variable	Description
$\Phi^{k}_{d,v,i}$	Indicates if the i^{th} instance of the VNF type $v \in V$ of the DC $d \in N_{dc}$ has been assigned to the slice $k \in K$.
$\Phi^{k,c}_{d,v,i}$	Indicates if the core network VNF $c \in N^{k,v}_{core}$ of type $v \in V$ of slice $k \in K$ has been mapped to the i^{th} instance of the same VNF type of the DC $d \in N_{dc}$.
$\Phi^{k,c,c^\star}_{d,v,i}$	Indicates if the VNFs $c^\star, c \in N^{k,v}_{core}$ ($c \neq c^\star$) of type $v \in V$ requested by the slice $k \in K$ are using the same i^{th} instance of the VNF v of the DC d assigned to the slice k.
$\Phi^{k,r}_{g}$	Indicates if the gNB $r \in N^{k}_{ran}$ of the slice $k \in K$ has been mapped to the substrate gNB $g \in N_{gnb}$.
$\Phi^{k,e'}_{e}, \zeta^{k,e'}_{e}$	Indicates if the virtual link $e' \in E^{k}_{slc}$ of the slice $k \in K$ has been mapped to the substrate link $e \in E_{dc}$, while $\zeta^{k,e'}_{e}$ represents data transmission time of e' over substrate link e.
ζ_e	Represents the data transmission time over the substrate link e.

The objective function of this MILP formulation is as follows:

$$min \sum_{d\in N_{dc}} \sum_{v\in V} \sum_{i\in N^{d,v}_{ist}} \sum_{k\in K} \sum_{\gamma\in\Gamma} \Upsilon_\gamma \Lambda_\gamma(d) \omega^v_\gamma \Phi^k_{d,v,i} + \\ + \sum_{k\in K} \sum_{e'\in E^k_{slc}} \sum_{e\in E_{dc}} \Upsilon_{bwt} \Lambda_{bwt} \omega_{bwt}(e') \Phi^{k,e'}_e + L \quad (4)$$

where Υ_γ and Υ_{bwt} are binary weighting factors, while $\Lambda_\gamma(d)$ and Λ_{bwt} are, γ resource usage cost on the DC $d \in N_{dc}$ and the cost per Mbps link bandwidth, respectively, with the latter being significantly cheaper. Three objective functions have been considered by varying Υ_γ and Υ_{bwt}. The first objective function (*MILP-Bwt*) seeks to minimize the FH/BH bandwidth consumption by selecting the weighting factors $\Upsilon_\gamma = 0$ and $\Upsilon_{bwt} = 1$. The second objective function (*MILP-Cost*) in which $\Upsilon_\gamma = \Upsilon_{bwt} = 1$ aims to minimize the E2E slice embedding cost. Lastly, the goal of the third objective function (*MILP-Mig*) is to minimize the number of VNF migrations and their effect onto the slices. *MILP-Mig* uses the same weighting factors of *MILP-Cost*. In contrast to *MILP-Cost*, however, in the case of *MILP-Mig*, γ resource usage cost $\Lambda_\gamma(d)$ depends also on the slice type apart from the DC itself. In *MILP-Mig*, if a VNF has to be migrated in order to meet the UP/CP latency demands of the requests then its effect onto already embedded slices is minimized by migrating the least utilized VNF.

All the aforementioned objective functions follow a dynamic slice embedding strategy. In essence, this means that with the arrival of a new E2E slice request, all the previously embedded requests along with the new one are re-embedded. This approach, although results in globally optimal embedding solutions for all the requests, may lead to possible VNF migrations that might entail service quality degradation for the users using the services provided by the slice owner. In order to tackle this problem, we also propose a static embedding strategy (*MILP-Mig-St*), which follows the same objective of *MILP-Mig* while embeds only the new slice request without affecting the already embedded slices.

We will now detail the constraints used in these MILP formulations. Regardless of the objective function, all the following constraints have to be satisfied in order for a solution to be valid. Each gNB $r \in N^k_{ran}$ in the slice $k \in K$ has to be mapped only on a single substrate gNB (Constraint (5)), which has to belong to the candidate set of r (Constraint (6)):

$$\sum_{g\in N_{gnb}} \Phi^{k,r}_g = 1 \quad \forall k \in K, \quad \forall r \in N^k_{ran} \quad (5)$$

$$\sum_{g\in N_{gnb}\setminus\Omega(r)} \Phi^{k,r}_g = 0 \quad \forall k \in K, \quad \forall r \in N^k_{ran} \quad (6)$$

Constraint (7) checks if the i^{th} instance of the VNF v of the DC d has been allocated to the E2E slice k, while Constraint (8) uses this information to make sure that the i^{th} instance of the VNF v is employed only by a single slice, thus guaranteeing isolation between slices.

$$\sum_{c\in N^{k,v}_{core}} \Phi^{k,c}_{d,v,i} - \mu_b \Phi^k_{d,v,i} \leq 0 \quad (7)$$

$$\forall k \in K, \quad \forall d \in N_{dc}, \quad \forall v \in V, \quad \forall i \in N^{d,v}_{ist}$$

$$\sum_{k\in K} \Phi^k_{d,v,i} \leq 1 \quad \forall d \in N_{dc}, \quad \forall v \in V, \quad \forall i \in N^{d,v}_{ist} \quad (8)$$

Constraint (9) ensures that the VNF c of type v of each gNB r in slice k is associated to a single VNF instance of its type.

$$\sum_{d\in N_{dc}} \sum_{i\in N^{d,v}_{ist}} \Phi^{k,c}_{d,v,i} = 1 \quad (9)$$

$$\forall k \in K, \quad \forall r \in N^k_{ran}, \quad \forall v \in V, \quad c = N^{k,v}_{core}(r)$$

Constraint (10) enforces for each virtual link there will be a continuous path between the gNB hosting the virtual gNBs and the DC(s) hosting the VNFs in the request. $E^{\star i}_{dc}$ is the set of the links that originate from any DC and directly arrive at the DC $i \in N_{dc}$, while $E^{i\star}_{dc}$ is the set of links that originates from the DC i and arrive at any DC directly connected to i.

$$\sum_{e\in E^{\star i}_{dc}} \Phi^{e^{n,m}}_e - \sum_{e\in E^{i\star}_{dc}} \Phi^{e^{n,m}}_e = \begin{cases} -1 & \text{if } i = n \\ 1 & \text{if } i = m \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

$$\forall i \in N_{dc}, \quad \forall e^{n,m} \in E_{slc}$$

Substrate gNBs can host virtual gNBs as long as they have sufficient capacity to satisfy their requested traffic demand:

$$\sum_{k\in K} \sum_{r\in N^k_{ran}} \omega_{tr}(r) \Phi^{k,r}_g \leq \omega_{tr}(g) \quad g \in N_{gnb} \quad (11)$$

Similarly, virtual links can be mapped onto a substrate link as long as the substrate link has sufficient capacity:

$$\sum_{k\in K} \sum_{e'\in E^k_{slc}} \omega_{bwt}(e') \Phi^{k,e'}_e \leq \omega_{bwt}(e) \quad \forall e \in E_{dc} \quad (12)$$

Each instance $i \in N^{d,v}_{ist}$ of the VNF type $v \in V$ on the DC $d \in N_{dc}$ can support maximum of $\omega^v_{gnb}(d,i)$ number of gNBs (Constraint 13), while ensuring that the capacity of $\gamma \in \Gamma$ resource type is not exceeded (Constraint 14):

$$\sum_{c\in N^{k,v}_{core}} \Phi^{k,c}_{d,v,i} \leq \omega^v_{gnb}(d,i) \quad (13)$$

$$\forall k \in K, \quad \forall d \in N_{dc}, \quad \forall v \in V, \quad \forall i \in N^{d,v}_{ist}$$

$$\sum_{c \in N_{core}^{k,v}} \omega_{\gamma}^{v}(c) \Phi_{d,v,i}^{k,c} \leq \omega_{\gamma}^{v}(d,i) \tag{14}$$

$$\forall k \in K, \quad \forall d \in N_{dc}, \quad \forall v \in V, \quad \forall i \in N_{ist}^{d,v}, \quad \forall \gamma \in \Gamma$$

Constraint (15) instead guarantees that a VNF can be spawned/instantiated on the DC $d \in N_{dc}$ as long as it has sufficient resources of type $\gamma \in \Gamma$ required to spawn/instantiate the requested VNF $v \in V$:

$$\sum_{k \in K} \sum_{v \in V} \sum_{i \in N_{ist}^{d,v}} \omega_{\gamma}^{v}(d,i) \Phi_{d,v,i}^{k} \leq \omega_{\gamma}(d) \quad \forall d \in N_{dc}, \quad \forall \gamma \in \Gamma \tag{15}$$

The transmission time ζ_e over the substrate link $e \in E_{dc}$ is computed by Constraint (16) based on the aggregated data demand on that link.

$$\sum_{k \in K} \sum_{e' \in E_{slc}^{k}} \frac{\omega_{bwt}(e')}{\omega_{bwt}(e)} \Phi_{e}^{k,e'} - \zeta_e = 0 \quad \forall e \in E_{dc} \tag{16}$$

Constraint (17) handles the accurate transmission time computation of the data on the virtual link e'.

$$\mu_b \Phi_{e}^{k,e'} + \zeta_e - \zeta_{e}^{k,e'} \leq \mu_b \quad \forall k \in K,\ e \in E_{dc},\ e' \in E_{slc}^{k} \tag{17}$$

Regardless of how much is the bandwidth requirement of virtual link $e' \in E_{slc}^{k}$, if this link has been mapped onto the substrate link $e \in E_{dc}$ ($\Phi_{e}^{k,e'} = 1$) then its transmission time is $\zeta_{e}^{k,e'} = \zeta_e$, which is computed considering the entire bandwidth demand on the substrate link e, as shown by Constraint (16). Note that the possibility of having $\zeta_{e}^{k,e'} = 1$ and $\Phi_{e}^{k,e'} = 0$ is ruled out since $\zeta_{e}^{k,e'}$ variable has a small positive coefficient μ_s (see formula (2)) in the considered objective function, which seeks to minimize the defined costs.

Finally, Constraint (18) and Constraint (19) ensure that the UP latency and the CP latency experienced by each gNB $r \in N_{ran}^{k}$ does not exceed the maximum acceptable UP latency ($\tau_{up}^{k}(r)$) and CP latency ($\tau_{cp}^{k}(r)$), respectively:

$$\sum_{c^{\star} \in N_{core}^{k,v}} \sum_{d \in N_{dc}} \sum_{i \in N_{ist}^{d,v}} \tau_{exc}^{up}(i, c^{\star}) \Phi_{d,v,i}^{k,c,c^{\star}} + \sum_{e \in E_{dc}} \zeta_{e}^{k,e'} \leq \tau_{up}^{k}(r) \tag{18}$$

$$\forall k \in K, \forall r \in N_{ran}^{k}, e' = \underline{E}_{slc}^{k,r}, v = V_{UPF}, c = N_{core}^{k,v}(r)$$

$$\sum_{d \in N_{dc}} \sum_{v \in V \setminus V_{UPF}} \sum_{i \in N_{ist}^{d,v}} \sum_{c \in N_{core}^{k,v}} \sum_{c^{\star} \in N_{core}^{k,v}} \tau_{exc}^{cp}(i, c^{\star}) \Phi_{d,v,i}^{k,c,c^{\star}} + + \sum_{e \in E_{dc}} \sum_{e' \in \overline{E}_{slc}^{k,r}} \tau_{tx}^{prop}(e) \Phi_{e}^{k,e'} \leq \tau_{cp}^{k}(r) \tag{19}$$

$$\forall k \in K, \quad \forall r \in N_{ran}^{k}$$

B. Heuristic

The MILP-based formulations become computationally intractable as the problem increases in size e.g., the size of the mobile network and/or the slice requests increase. For example, the MILP–based dynamic placement algorithms take a day on Intel Core i7 laptop (3.0 GHz CPU, 16 Gb RAM) using the ILOG CPLEX 12.8 solver to embed 10 slice requests each composed of 10 gNBs in the substrate network composed of 6 L1 DCs, 2 L2 DCs and a single L3 DC. In order to address this scalability issue, we developed a dynamic placement heuristic, shown in Algorithm 1, which is able to embed the same requests in less than a second. The proposed heuristic pursues the objective of minimizing the number of migrations of the VNF instances and their effect onto other slices, which is achieved in four steps. In the first step, the heuristic initially sorts the slice requests in ascending order according to their CP/UP latency QoS requirements per gNB. Within the requests with same QoS requirements, the slices are then sorted in descending order according to the quantity of the requested gNBs per slice. In the second step, candidate substrate gNBs are selected for each gNB per slice request starting from the first one in the ordered list based on the desired location of the requested gNBs and resource availability at the candidate gNB.

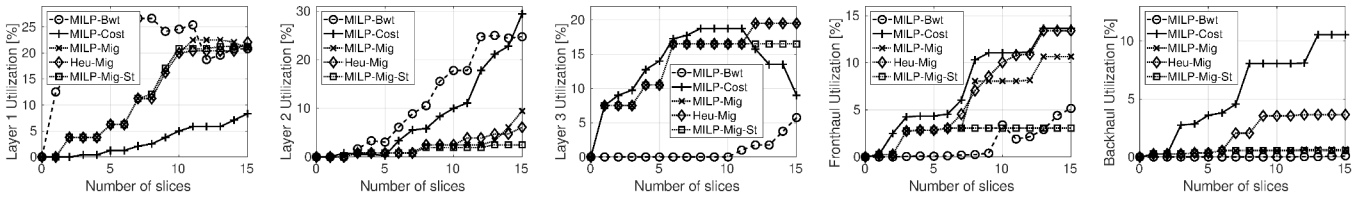
Algorithm 1: *Heuristic (Heu-Mig)*

Data: (G_{dc}, G_{slc})
Result: Slice placement and resource allocation.
Step 1: Ordering of slices;
- *Order slices according to their CP/UP latency QoS requirements* $\uparrow$;
- *Order slices having the same QoS according to # of gNB requests* $\downarrow$;

Step 2: Candidate selection for gNBs in slice requests;
for $k \in K \uparrow$ **do**
  for $r \in N_{ran}^{k}$ **do**
    $cand\_gnb\{k\}(r) \leftarrow \emptyset$;
    for $g \in N_{gnb}$ **do**
      $dist \leftarrow dis(loc(r), loc(g))$;
      if $dist \leq \delta(g)$ ***and*** $\omega_{tr}(r) \leq \omega_{tr}(g)$ **then**
        $cand\_gnb\{k\}(r) \leftarrow g$;

Step 3: Estimate VNF hosting cost on each DC;
for $d \in N_{dc}$ **do**
  $C_{map}(d) \leftarrow C_{dc}(QoS(k), d)$;
  for $r \in N_{ran}^{k}$ **do**
    $C_{curr} \leftarrow +\infty$;
    for $h \in cand\_gnb\{k\}(r)$ **do**
      $C_{new} \leftarrow C_{link}(d, h)$;
      $C_{curr} \leftarrow min(C_{curr}, C_{new})$;
    $C_{map}(d) \leftarrow C_{map}(d) + C_{curr}$;

Step 4: Perform gNB mapping, DC selection and resource allocation;
$d \leftarrow argmin(C_{map})$;
for $r \in N_{ran}^{k}$ **do**
  for $v \in V$ **do**
    $flag \leftarrow 0$;
    while $flag = 0$ ***or*** $argmin(C_{map}) \neq +\infty$ **do**
      if $\omega_{\gamma}^{v} \leq \omega_{\gamma}(d)\ \forall\ \gamma \in \Gamma$ **then**
        $vnf\_host \leftarrow d$;
        $C_{curr} \leftarrow +\infty$;
        $flag \leftarrow 1$;
        if *v is* UPF **then**
          for $h \in cand\_gnb\{k\}(r)$ **do**
            if *UP laty* $\leq \tau_{up}^{k}(\forall \hat{r} \in N_{ran}^{k})$ **then**
              $C_{new} \leftarrow C_{link}(h, d)$;
              $C_{curr} \leftarrow min(C_{curr}, C_{new})$;
              $best\_gnb \leftarrow h$;
            else
              $flag \leftarrow 0$;
          $mapped\_ran(r) \leftarrow best\_gnb$;
        else if *CP laty* $\geq \tau_{cp}^{k}(\forall \hat{r} \in N_{ran}^{k})$ **then**
          $flag \leftarrow 0$;
          $C_{map}(d) \leftarrow +\infty$;
          $d \leftarrow argmin(C_{map})$;
      else
        $C_{map}(d) \leftarrow +\infty$;
        $d \leftarrow argmin(C_{map})$;
    $mapped\_core(v) \leftarrow vnf\_host$;
    if *flag = 1* **then**
      - *Assign path*;
      - *Allocate and update network resources*;
      - *Update UP/CP latency budget* ($\tau_{up,cp}^{k}$);



(a) Utilization of L1 DCs. (b) Utilization of L2 DCs. (c) Utilization of L3 DCs. (d) Fronthaul utilization. (e) Backhaul utilization.

Fig. 2: Utilization of Layer 1, Layer 2 and Layer 3 DCs along with the utilization of FH and BH links.

This is followed by estimating the cost of hosting VNFs per slice request on each DC in the third step. This encompasses both the VNF instantiation cost as well as the FH/BH link usage cost. The former depends on QoS of the requested slice as well as on the DC type e.g., L1, L2 or L3; for instance, for the requests with tight UP/CP latency requirements (e.g., URLLC slices), it is cheaper to place VNFs on the L1 DCs, while in the case of loose UP/CP latency requirements (e.g., mMTC), it is cheaper to place VNFs on the L3 DCs. The latter instead depends on the host DC, gNB, and traffic demand.

Based on the VNF mapping cost computed for each DC in the previous step, in the step four, the heuristic seeks to find the cheapest host gNB and DC, where the latter has the required amount of resources for the VNF to be spawned/instantiated and would not violate the UP/CP latency requirement of any gNB in all the slice requests that have already been embedded. This is followed by performing the gNB and the VNFs mappings onto, respectively, the substrate gNB and the DC that would yield the cheapest embedding cost. The heuristic then assigns both the CP and the UP paths between the nodes in the slice request using Dijkstra's shortest path algorithm, allocating the required amount of node and link resources and updating the network resources. These steps are repeated for all the slice requests. The overall time complexity of this heuristic algorithm is $O(k[\log k + ne \log n(m_r + 1) + m_r n_r(n+1)])$, where k, m_r, n_r, n, and e are, respectively, the number of slices, gNBs per slice, the gNBs in the substrate network, the DCs in all layers, and the substrate links.

V. Evaluation

The goal of this section is to compare the *relative* performances of the MILP-based dynamic/static placement algorithms and of the dynamic placement heuristic algorithm.

A. Simulation Environment

The mobile network considered in the simulations is composed of 7 nodes, similar to the one depicted in Figure 1a. The L3 DC is connected to the L2 DCs by means of 2Gbps of BH links, while the L2 DCs are connected to the L1 DCs by means of 1Gbps of FH links. L1, L2 and L3 DCs have, respectively, 6, 18 and 36 vCPUs, 12, 36 and 72 GB vRAM and 24, 72 and 144 GB vSTO. For the sake of simplicity, it is assumed that 1 vCPU, 2GB vRAM and 4GB vSTO are required to instantiate/spawn a VNF (e.g., AMF, SMF, UPF). The capacity of the CP VNFs is expressed in terms of the number of CP events they can handle per hour, which are 125000 $[ev/h]$ for an AMF instance and 250000 $[ev/h]$ for an SMF instance, following the approach presented in [43]. Whereas, the UP VNF instance (UPF) capacity is characterized by its maximum supportable traffic and is 7.5Gbps [46]. Additionally, due to the scalability issue of the dynamic MILP-based algorithms, we limit the number of gNBs that can use the same VNF instance to 10, which is the maximum number of gNBs that a slice can request (see Table IV).

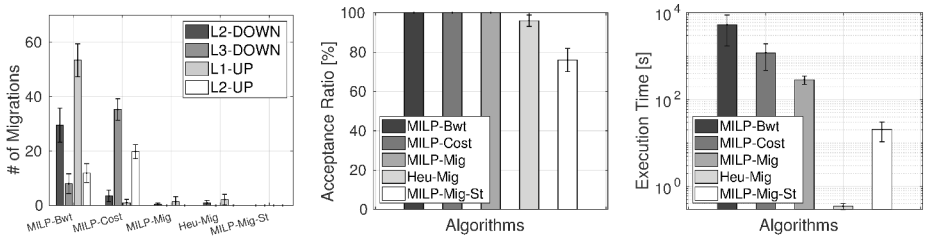
TABLE IV: Types and parameters of E2E slice requests.

Slice type	CP/UP lat.	CP events/gNB	Data rate/gNB	# of gNBs
eMBB	20/100 ms	500 $[ev/h]$	100 Mbps	$1-2$
URLLC	5/25 ms	750 $[ev/h]$	10 Mbps	$2-6$
mMTC	60/300 ms	1000 $[ev/h]$	1 Mbps	$6-10$

Three types of network slices are considered in the simulations each of which is characterized by its CP/UP latency, data rate and the requested number of gNBs as summarized Table IV. Note that while the reported parameter values do not reflect all types of novel applications and services that 5G technology is expected to support, it captures the general characteristics of all slice types. For example, eMBB slices are characterized by the highest bandwidth requirements, the URLLC slices are characterized by the strictest UP/CP latency, while the mMTC slices have the gNB requirements with the highest density in order to support low data rate communications between sensors and actuators in the Internet of Things (IoT) scenarios, for example. In the simulations, the slice requests are randomly picked within the mentioned slice types and are mapped sequentially onto the substrate network. The reported results are the average of 10 simulation runs, each of which, due to the scalability issue of the MILP-based algorithm, is composed of 15 slice requests.

B. Simulation Results

DC utilization. Since an upper bound is set on the number of gNBs in the slice requests that can share the same VNF instance, the capacity of each DC is expressed on the total number of gNBs that can use the VNFs deployed on that DC. Figure 2a, Fig. 2b and Fig. 2c show the utilization of the DCs, respectively, on L1, L2 and L3 for all the algorithms described in Section IV for a single simulation run (15 slice requests). It can be observed that for the *MILP-Bwt* algorithm, while the DC utilization starts increasing dramatically at L1 right from the 1^{th} slice embedding, the VNFs start to be placed at L3 after the 10^{th} slice embedding. This is expected since the objective of this algorithm is to minimize the bandwidth consumption in the networks, which results in embedding the E2E slice requests as closer to the gNBs as possible. The trend is reversed for the *MILP-Cost* algorithm that seeks to minimize the embedding cost, assuming that the link resource is much cheaper than the node resource (e.g., vCPU, vRAM, vSTO). As a consequence, this approach starts instantiating VNFs from L3, which yields the cheapest embedding cost due



(a) VNF migrations. (b) Acceptance ratio. (c) Execution time.

Fig. 3: Number of VNF migrations, acceptance ratio and execution time for all algorithms.

to plenty of available resources on L3 DCs. As for the rest of the algorithms that aim to minimize the number of VNF migrations as well as the effect of the migrations onto the network, it can be observed that their DC utilization resembles each other following a non-decreasing trend as opposed to the *MILP-Bwt* and *MILP-Cost* algorithms. This is because the DC utilization for the *MILP-Bwt* and *MILP-Cost* algorithms plummets at different layers due to VNF migrations that are much higher than for the other algorithms as we will see in Fig. 3a. Moreover, it can be observed that *MILP-Mig*, *Heu-Mig* and *MILP-Mig-St* algorithms, being aware of requested E2E slice type (e.g., URLLC, eMBB, mMTC), utilize the DC resources more efficiently at all layers, avoiding their over- and under-utilization.

Link utilization. Figure 2d and Fig. 2e illustrate, respectively, the FH and BH link utilization as a function of the number E2E slice requests for the same simulation run. As expected, the *MILP-Bwt* and *MILP-Cost* algorithms achieve, respectively, the lowest and the highest level of FH and BH link utilization. As for the rest of the algorithms, we can observe that their link utilization lies somewhere in between the ones of *MILP-Bwt* and *MILP-Cost*. This is due to the fact that the VNF migration minimization strategy, which is implemented by *MILP-Mig*, *Heu-Mig* and *MILP-Mig-St* algorithms, makes embedding decisions by taking into account also the E2E slice type apart from their requirements. It is worthwhile to observe that while the L3 DC utilization of *MILP-Bwt* reaches more than 5% after mapping 15 slice requests (see Fig. 2c), its corresponding BH utilization is less than 1% (see Fig. 2e). This is because MILP-Bwt aims at minimizing the bandwidth utilization, which is achieved by mapping only the control plane VNFs (AMF, SMF) onto the L3 DC since they have negligible traffic demand compared to that of the user plane VNF (UPF).

Number of VNF migrations. The average number of VNF migrations in different inter-layer directions, the acceptance ratio, and execution time with 95% confidence intervals are shown in Fig. 3. It can be seen that *MILP-Bwt* yields the highest aggregated number of VNF migrations in which the migrations from L1 DCs towards the upper layer DCs constitute the major share of VNF migrations (see Fig. 3a). This is a result of scarce resources at L1 DCs, which at some point results in a number of VNF migrations in order to embed the E2E slice requests that have more stringent CP/UP latency requirements (i.e., URLLC slices). The second highest aggregated number of VNF migrations takes place in the case of employing *MILP-Cost* with the L3 DCs towards the lower layer DC having the major share. This is justified by the fact that at some point due to high FH/BH transmission time, resulted by increased aggregated data demand over the links, the algorithm migrates the VNFs of the URLLC slice requests from the L3 DC in order for their CP/UP latency requirements to be satisfied. For what concerns the migration minimization algorithms, they achieve much lower VNF migrations with the *MILP-Mig-St* being zero due to its static placement nature.

Acceptance ratio. Figure 3b displays the acceptance ratio of slice requests for all the considered algorithms. As expected, all the dynamic placement MILP-based algorithms (i.e., *MILP-Bwt*, *MILP-Cost*, and *MILP-Mig*) accept equal number of requests. As opposed to *MILP-Mig* counterpart, which accepts all the requests, *Heu-Mig* accepts slightly less number of requests (around 95%), while the static placement *MILP-Mig-St* algorithm rejects more than 20% of requests on average. This is because after each slice embedding, *MILP-Mig-St* considers updated substrate network for the subsequent slice request, which confines its embedding possibilities.

Execution time. Finally, Fig. 3c shows the average execution time for all algorithms. It is obvious that *Heu-Mig* is able to embed the slice requests in a significantly shorter time compared to the rest of the algorithms, proving its scalability.

VI. Conclusions

In this study, we proposed and compared different strategies for the E2E slice placement problem in the 5G network. Based on the reported results, we can conclude that *MILP-Bwt* is the most appropriate algorithm to be used in the network segment in which the transport network lacks capacity e.g., in the areas where trenching optical fiber connection is infeasible. Conversely, the *MILP-Cost* algorithm is more suitable to be used in the part of the mobile network where there is plenty of transport network resources, while the DCs closer to the users have a very limited amount of resources that have to be utilized smartly. Nonetheless, both of these algorithms result in a significant number of VNF migrations, which might lead to slice users' performance degradation. It has been demonstrated that using the *MILP-Mig* algorithm and its corresponding *Heu-Mig* heuristic, the number of VNF migrations can be notably curtailed while achieving a better compromise between the utilization of DCs and FH/BH links. As for the static placement algorithm (*MILP-Mig-St*), although it achieves no VNF migration, it causes around 20% slice request rejections, which undermines its efficiency.

REFERENCES

[1] G. P. A. W. Group *et al.*, "View on 5g architecture (version 2.0)," *White Paper, July*, 2017.

[2] M. Series, "Imt vision–framework and overall objectives of the future development of imt for 2020 and beyond," *Recommendation ITU*, pp. 2083–0, 2015.

[3] P. K. Agyapong, M. Iwamura, D. Staehle, W. Kiess, and A. Benjebbour, "Design considerations for a 5g network architecture." *IEEE Communications Magazine*, vol. 52, no. 11, pp. 65–75, 2014.

[4] P. Rost, A. Banchs, I. Berberana, M. Breitbach, M. Doll, H. Droste, C. Mannweiler, M. A. Puente, K. Samdanis, and B. Sayadi, "Mobile network architecture evolution toward 5g," *IEEE Communications Magazine*, vol. 54, no. 5, pp. 84–91, 2016.

[5] R. Su, D. Zhang, R. Venkatesan, Z. Gong, C. Li, F. Ding, F. Jiang, and Z. Zhu, "Resource allocation for network slicing in 5G telecommunication networks: A survey of principles and models," *IEEE Network*, 2019.

[6] X. Foukas, G. Patounas, A. Elmokashfi, and M. K. Marina, "Network slicing in 5g: Survey and challenges," *IEEE Communications Magazine*, vol. 55, no. 5, pp. 94–100, 2017.

[7] 3GPP, "System architecture for the 5g system," 3GPP TS 23.501 Version 15.5.0 Release 15, March 2019.

[8] ——, "Procedures for the 5g system," 3GPP TS 23.502 Version 15.1.0 Release 15, March 2018.

[9] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[10] B. Xiang, J. Elias, F. Martignon, and E. Di Nitto, "Joint network slicing and mobile edge computing in 5G networks," in *IEEE International Conference on Communications (ICC)*. IEEE, 2019, pp. 1–7.

[11] K. Chen and R. Duan, "C-RAN the road towards green RAN," *China Mobile Research Institute, white paper*, vol. 2, 2011.

[12] S. Zhang, "An overview of network slicing for 5G," *IEEE Wireless Communications*, 2019.

[13] F. Z. Yousaf, J. Lessmann, P. Loureiro, and S. Schmid, "Softepc - dynamic instantiation of mobile core network entities for efficient resource utilization," in *Proc. of IEEE ICC*, Budapest, Hungary, 2013.

[14] M. Bagaa, T. Taleb, A. Laghrissi, and A. Ksentini, "Efficient virtual evolved packet core deployment across multiple cloud domains," in *Proc. of IEEE WCNC*, Barcelona, Spain, 2018.

[15] R. Addad, M. Bagaa, T. Taleb, D. L. C. Dutra, and H. Flinck, "Optimization model for cross-domain network slices in 5g networks," *IEEE Transactions on Mobile Computing*, 2019.

[16] T.-X. Do and Y. Kim, "Latency-aware Placement for State Management Functions in Service-based 5G Mobile Core Network," in *Proc. of IEEE ICCE*, Hue, Vietnam, 2018.

[17] D. Harutyunyan and R. Riggio, "Flex5G: Flexible Functional Split in 5G Networks," *IEEE Transactions on Network and Service Management*, vol. 15, no. 3, pp. 961–975, 2018.

[18] M. I. Kamel, L. B. Le, and A. Girard, "LTE wireless network virtualization: Dynamic slicing via flexible scheduling," in *Proc. of IEEE VTC2014-Fall*, Vancouver, Canada, 2014.

[19] M. Richart, J. Baliosian, J. Serrat, and J.-L. Gorricho, "Resource slicing in virtual wireless networks: A survey," *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 462–476, 2016.

[20] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, "Network slicing and softwarization: A survey on principles, enabling technologies, and solutions," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2429–2453, 2018.

[21] Y. Zaki, L. Zhao, C. Goerg, and A. Timm-Giel, "LTE wireless virtualization and spectrum management," in *Proc. of IFIP WMNC*, Budapest, Hungary, 2010.

[22] R. Mahindra, M. A. A. Khojastepour, H. Zhang, and S. Rangarajan, "Network-wide radio access network sharing in cellular networks," *Proc. IEEE ICNP, Goettingen, Germany*, pp. 1–10, 2013.

[23] J. He and W. Song, "AppRAN: Application-oriented radio access network sharing in mobile networks," in *Proc. of IEEE ICC*, London, UK, 2015.

[24] A. Ksentini and N. Nikaein, "Toward enforcing network slicing on ran: Flexibility and resources abstraction," *IEEE Communications Magazine*, vol. 55, no. 6, pp. 102–108, 2017.

[25] O. Sallent, J. Perez-Romero, R. Ferrus, and R. Agusti, "On radio access network slicing from a radio resource management perspective," *IEEE Wireless Communications*, vol. 24, no. 5, pp. 166–174, 2017.

[26] M. Yang, Y. Li, L. Zeng, D. Jin, and L. Su, "Karnaugh-map like online embedding algorithm of wireless virtualization," in *Proc. of IEEE WPMC*, Taipei, Taiwan, 2012.

[27] K. Pentikousis, Y. Wang, and W. Hu, "Mobileflow: Toward software-defined mobile networks," *IEEE Communications magazine*, vol. 51, no. 7, pp. 44–53, 2013.

[28] M. R. Sama, L. M. Contreras, J. Kaippallimalil, I. Akiyoshi, H. Qian, and H. Ni, "Software-defined control of the virtualized mobile packet core," *IEEE Communications Magazine*, vol. 53, no. 2, pp. 107–115, 2015.

[29] A. Baumgartner, V. S. Reddy, and T. Bauschert, "Combined virtual mobile core network function placement and topology optimization with latency bounds," in *Proc. of IEEE EWSDN*, Bilbao, Spain, 2015.

[30] M. Bagaa, T. Taleb, and A. Ksentini, "Service-aware network function placement for efficient traffic handling in carrier cloud," in *Proc. of IEEE WCNC*, Istanbul, Turkey, 2014.

[31] Z. A. Qazi, P. K. Penumarthi, V. Sekar, V. Gopalakrishnan, K. Joshi, and S. R. Das, "KLEIN: A minimally disruptive design for an elastic cellular core," in *Proc. of ACM SOSR*, Santa Clara, CA, 2016.

[32] A. Basta, W. Kellerer, M. Hoffmann, K. Hoffmann, and E.-D. Schmidt, "A virtual sdn-enabled lte epc architecture: A case study for s-/p-gateways functions," in *Proc. of IEEE SDN4FNS*, Trento, Italy, 2013.

[33] A. Basta, W. Kellerer, M. Hoffmann, H. J. Morper, and K. Hoffmann, "Applying NFV and SDN to LTE mobile core gateways, the functions placement problem," in *Proc. of ACM 4th workshop on All things cellular: operations, applications, & challenges*. London, UK, 2014.

[34] A. Kaloxylos, "A survey and an analysis of network slicing in 5g networks," *IEEE Communications Standards Magazine*, vol. 2, no. 1, pp. 60–65, 2018.

[35] W. Guan, X. Wen, L. Wang, Z. Lu, and Y. Shen, "A service-oriented deployment policy of end-to-end network slicing based on complex network theory," *IEEE Access*, vol. 6, pp. 19 691–19 701, 2018.

[36] Nakao, Akihiro and Du, Ping and Kiriha, Yoshiaki and Granelli, Fabrizio and Gebremariam, Anteneh Atumo and Taleb, Tarik and Bagaa, Miloud, "End-to-end network slicing for 5G mobile networks," *Journal of Information Processing*, vol. 25, pp. 153–163, 2017.

[37] I. Quintana-Ramirez, A. Tsiopoulos, M. A. Lema, F. Sardis, L. Sequeira, J. Arias, A. Raman, A. Azam, and M. Dohler, "The making of 5g: Building an end-to-end 5g-enabled system," *IEEE Communications Standards Magazine*, vol. 2, no. 4, pp. 88–96, 2018.

[38] X. An, C. Zhou, R. Trivisonno, R. Guerzoni, A. Kaloxylos, D. Soldani, and A. Hecker, "On end to end network slicing for 5g communication systems," *Transactions on Emerging Telecommunications Technologies*, vol. 28, no. 4, p. e3058, 2017.

[39] H. Halabian, "Distributed resource allocation optimization in 5g virtualized networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 3, pp. 627–642, 2019.

[40] D. Harutyunyan, S. Nashid, B. Raouf, and R. Riggio, "Latency–Aware Service Function Chain Placement in 5G Mobile Networks," in *Proc. of IEEE NetSoft*, Paris, France, 2019.

[41] Y. Q. Bian and D. Rao, "Small cells big opportunities," *Global Business Consulting. Huawei Technologies Co., Ltd*, 2014.

[42] D. Harutyunyan and R. Riggio, "How to migrate from operational lte/lte–a networks to c–ran with minimal investment?" *IEEE Transactions on Network and Service Management*, vol. 15, no. 4, pp. 1503–1515, 2018.

[43] F. Z. Yousaf, P. Loureiro, F. Zdarsky, T. Taleb, and M. Liebsch, "Cost analysis of initial deployment strategies for virtualized mobile core network functions," *IEEE Communications Magazine*, vol. 53, no. 12, pp. 60–66, Dec 2015.

[44] A. Fischer, J. F. Botero, M. T. Beck, H. De Meer, and X. Hesselbach, "Virtual network embedding: A survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 1888–1906, 2013.

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

[46] Intel, "Towards achieving high performance in 5g mobile packet cores user plane function," Intel, White Paper, June 2018.