Latency–Aware Service Function Chain Placement in 5G Mobile Networks

Davit Harutyunyan*, Nashid Shahriar§, Raouf Boutaba§ and Roberto Riggio*

*FBK CREATE-NET, Italy; Email: d.harutyunyan,rriggio@fbk.eu
§David R. Cheriton School of Computer Science, University of Waterloo, Canada; Email: nshahria,rboutaba@uwaterloo.ca

***Abstract*—The 5th generation mobile network (5G) is expected to support numerous services with versatile quality of service (QoS) requirements such as high data rates and low end-to-end (E2E) latency. It is widely agreed that E2E latency can be significantly reduced by moving content / computing capability closer to the network edge. However, since the edge nodes (i.e., base stations) have limited computing capacity, mobile network operators shall make a decision on how to provision the computing resources to the services in order to make sure that the E2E latency requirement of the services are satisfied while the network resources (e.g., computing, radio, and transport network resources) are used in an efficient manner.**

In this work, we employ integer linear programming (ILP) techniques to formulate and solve a joint user association, service function chain (SFC) placement, and resource allocation problem where SFCs, composed of virtualized service functions (VSFs), represent user requested services that have certain E2E latency and data rate requirements. Specifically, we compare three variants of an ILP-based algorithm that aim to minimize E2E latency of requested services, service provisioning cost, and VSF migration frequency, respectively. We then propose a heuristic in order to address the scalability issue of the ILP-based solutions. Simulations results demonstrate the effectiveness of the proposed heuristic algorithm.

***Index Terms*—Latency-sensitive Services, Resource Allocation, Service Function Chain Placement, Mobile Networks.**

I. Introduction

The 5th generation of mobile communication systems (5G) is on the horizon with the promise to revolutionize the communication landscape. 5G will enable a wide variety of services, including massive broadband, machine to machine communications, tactile Internet, virtual/augmented reality, high definition media delivery, autonomous vehicles, real-time monitoring and control, and so on [1]. Many of these services will have stringent quality of service (QoS) requirements in terms of data transmission rate, latency, jitter, reliability, and mobility [2], [3]. For instance, ultra-low latency services (e.g. virtual/augmented reality, real-time monitoring, and so on) urge data to be delivered satisfying strict end-to-end (E2E) latency budget and particular data transmission rate, whereas best-effort broadband communications have to provide gigabytes of bandwidth with no particular latency requirements.

To support such versatile and ambitious QoS requirements of different 5G services, mobile network infrastructure will need to undergo a paradigm shift towards adding distributed micro/edge data centers (DCs) [1]. For example, sub-milisecond latency services facilitating virtual/augmented reality may be composed of multiple service functions (SFs) some of which (e.g., video rendering) may need to be processed right at the decentralized units (DUs) collocated with antennas, thus avoiding the round-trip delay to and from either centralized units (CUs) or core DCs. Similarly, media delivery services with loose latency requirement may still cache bulky media contents at the DUs in order to avoid the bandwidth burden on the expensive fronthaul/backhaul (FH/BH) links. To cope with such requirements, mobile networks will have to equip DUs with additional computing resources, turning them into micro DCs, that incurs both capital expenditure (CapEx) and operational expenses (OpEx). Similarly, each CU, that serves user equipments (UEs) from multiple DUs, will have to be converted to light-weight DCs, also known as cloudlets. The core DCs will still be there providing abundance of computing resources at a cheaper cost than CUs and DUs.

Another enabling technology expected to play a key role in 5G is virtualization that decouples SFs from dedicated proprietary hardware and deploys virtualized service functions (VSFs) on commodity servers, thus reducing CapEx [4], [5]. Virtualization provides the opportunity to deploy VSFs at core DCs and cloudlets, or even at DUs, based on the QoS requirements and demands of services. Each of these services can be composed of different kinds and numbers of SFs that are interconnected in a particular order, also known as service functions chains (SFCs). An SFC can have its acceptable E2E latency budget and data rate requirement as per the UE's demand. In addition, a VSF has its own computing capacity demand that can be used to process data from a finite number of SFCs requiring the same SF. However, sharing a VSF among multiple SFCs may increase both processing time of the VSF and transmission delay at the physical machine where the VSF is hosted. Furthermore, VSF sharing among SFCs whose UEs are located in distant geographical regions may impose unnecessary burden on FH/BH links. On the other hand, it is impossible to instantiate a separate VSF for each UE due to the finite computing capacity and link bandwidth available at DU, CU, and core DCs and FH/BH links, respectively. Therefore, instantiating the optimal number of VSFs in different DCs of a mobile network and associating them to UEs even for a known set of SFCs is a non-trivial problem.

The UE association, SFC placement and resource allocation problem is further complicated by the skewness in the amount

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).

978-1-5386-9376-6/19/$31.00

of computing resources at different DCs and the existence of heterogeneous services with distinct QoS requirements. Since the number of DUs is large and they are distributed in remote geographic locations, the amount of computing resources in DUs will be very limited. An SFC placement strategy aiming to minimize E2E latency for all the SFC requests can prefer to place VSFs to DUs regardless of the QoS requirement, thus exhausting computing resources of DUs in no time. This strategy will need to migrate VSFs whose SFCs do not require strict latency from DUs to CUs or to core DCs in order to accommodate newly arrived SFCs with strict latency requirements. Similarly, another strategy that initially places VSFs of SFCs in core DCs irrespective of QoS requirements needs to adjust VSF placement later on. For instance, a VSF placed in a core DC could satisfy strict latency requirement when there is a light load and starts to violate its latency constraint as the load increases due to increase in transmission and processing delays along the other VSFs of the SFC. This strategy will also result in an increased number of migrations in order to help the violated SFCs satisfy latency constraints. Therefore, a sought-after SFC placement strategy should minimize migration frequencies as migration causes disruption of services.

In this paper, we demonstrate the pros and cons of different SFC placement strategies through empirical simulation of a 5G mobile network. To do so, we employ integer linear programming (ILP) techniques to formulate and solve a joint UE association, SFC placement, and resource allocation problem, where SFCs represent services with certain E2E latency and data rate requirements requested by UEs located in different areas of the mobile network. Specifically, we compare three variants of the ILP formulation that aim to minimize E2E latency of requested services, service provisioning cost, and VSF migration frequency, respectively. We also develop a comprehensive E2E latency model suitable for SFCs in the 5G mobile networks. We then propose a heuristic in order to address scalability issue of the ILP formulation.

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 SFC request models are introduced in Sec. III. The ILP 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

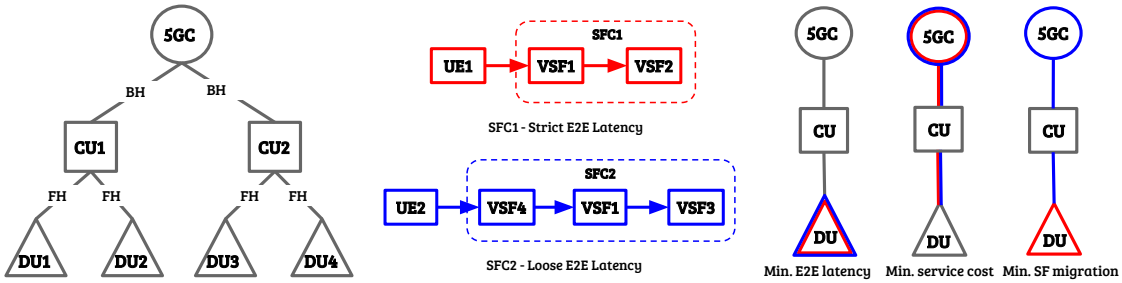
One of the problems tackled in our study is the server selection in heterogeneous cloud network for computation offloading. There is a sizable body of work published on this problem [6], [7], [8]. A hierarchical edge cloud architecture is proposed in [6]. The main idea is to offload users' computational tasks to the clouds preferably closer to the users in order to reduce their task completion time. The authors of [7] propose a heuristic local/remote cloud server selection algorithm that aims to increase the probability of successfully executing the tasks within their delay constraints. Another server selection problem is presented in [8]. Initially, users are grouped into clusters where users belonging to the same cluster have similar latency to remote servers. The clustered users' demand is then assigned to the appropriate servers with the goal of minimizing the overall latency by shortening the distance between clusters and servers. However, none of the aforementioned studies consider realistic latency-sensitive applications with actual E2E latency requirement envisioned to be supported in 5G networks. Moreover, they use simplistic models for latency estimation, neglecting many sources of latency present in real-life mobile networks.

Another thrust of research, relevant to our study, targets the SFC placement problem, having a certain E2E latency requirement to be satisfied [9], [10], [11], [12]. A delay-aware SFC placement problem is studied in [9]. The main idea is to place VSFs composing SFCs in a way as to satisfy the E2E latency of the requested services while using the network resources efficiently. [10] studies the joint VSF placement and CPU allocation problem in 5G networks. Employing a queuing-based system model, an optimization problem is formulated seeking to minimize the ratio between the actual and the maximum allowed latency, across all services. The VSF instantiation and migration problem is studied in [11], having the goal of maximizing network throughput by dynamically admitting as many requests as possible, while ensuring that their resource demands and E2E latency requirements are satisfied. The authors of [12] study the same problem with the goal of minimizing SFC delays. An MILP model is employed to decide whether to re-instantiate or migrate the VSFs and find their optimal placements while seeking to achieve minimal downtimes for the VSFs. However, for these studies, the E2E latency computation is confined within SFCs, disregarding the baseband processing time along with the transmission/propagation time over the air interface. Moreover, the authors do not consider heterogeneous servers, which augment the search space, making the SFC placement problem more cumbersome.

The closest related work to ours are [13] and [14]. [13] selects an ordered sequence of VSFs and data delivery paths connecting them to establish an SFC while minimizing overall latency. On the other hand, [14] formulates the delay-aware VSF scheduling and network resource allocation for SFCs by considering both VSFs processing delays and SFC transmission delays at virtual links. However, both of these works consider VSF instances already being placed and ignore capacity constraints in the nodes. In contrast, we consider the joint problem of SFC placement, user association, and network resource allocation that allows optimization of both computing and network resources based on users' location, services' demands and QoS requirements. Several other models, including [15], [16], [17], have been proposed for quantifying E2E latency in the context of virtual network. Our proposed latency model stands out these models in considering delays in the context of 5G mobile network including, delay in the air interface and VSF processing delays as a function of the number of users sharing the air interface and computing resource, respectively.

III. Network Model

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



(a) Mobile network topology (b) SFC requests. (c) SFC placement.

Fig. 1: Sample mobile network, SFC requests and SFC placements.

A. Problem Statement

Figure 1a depicts the reference network architecture for the joint UE association, SFC placement and resource allocation problem. Let us consider a 5G network with the NG–RAN architecture [18] in which we assume that, like traditional evolved NodeBs, DUs can still perform all baseband signal processing. While each CU can serve multiple DUs over FH links, each core network (5GC) can serve multiple CUs over BH links. We assume that in this hierarchical network architecture, the closer is the DC to the 5GC, the more is its computing capacity. Figure 1b illustrates examples of SFCs with the red and blue requests having, respectively, strict and loose latency requirements. Receiving SFC requests by the UEs, the network provider shall associate UEs to DUs and place the requested SFCs allocating sufficient resources (e,g., physical resource blocks (PRBs), computing resource (CPU), FH/BH bandwidth) 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, there may be several mapping options each minimizing a certain cost function. The problem of UE association, latency-aware SFC placement and resource allocation can be formally stated as follows:

Given: a 5G network with the NG–RAN architecture, the computing capacity of each DC/node (e.g., DU, CU, core), the transport network topology with the capacity of each FH/BH link and UEs with their requested SFCs along with the data rate and E2E latency requirements of the requested services.

Find: UEs associations, SFC placements and resource allocation in the network.

Objectives: (i) minimize E2E latency for all UEs, (ii) minimize the overall service provisioning cost and (iii) minimize the migration frequency of VSFs.

B. Mobile Network Model

Let $G_{net} = (N_{net}, E_{net})$ be an *undirected* graph modelling the mobile network, where $N_{net} = N_{du} \cup N_{cu} \cup N_{core}$ is the union of the set of DUs, CUs and the core node. E_{net} is the set of FH and BH links. An edge $e^{nm} \in E_{net}$ exists if and only if a connection exists between $n, m \in N_{net}$. Each network node $n \in N_{net}$ has $\omega_{cpu}(n)$ computing capacity expressed in terms of the number of CPUs, and a single CPU is required per VSF to be instantiated. Each VSF $s \in N_{vsf}$ instantiated at the node $n \in N_{net}$, has capacity $\omega^s_{num}(n)$ expressed in terms of the maximum number of UEs that can share the same VSF mapped on the node. It is worth mentioning that the model also tackles the case where, due to high VSF demand, multiple instances of the same VSF are needed. Each node $n \in N_{net}$ is associated with a geographic location $loc(n)$, as x, y coordinates while each DU $d \in N_{du}$ is also associated with a coverage radius of $\delta(d)$, in meters. It is assumed that DUs have sufficient amount of PRBs in order to meet the data rate demand of the requested services, especially in a small-cell deployment scenario that enables aggressive frequency reuse. Another weight $\omega_{bwt}(e^{nm})$ is assigned to each link $e^{nm} \in E_{net} : \omega_{bwt}(e^{nm}) \in \mathbb{N}^+$ representing the capacity (in Gbps) of the FH/BH link connecting the nodes n and m. Table I summarizes the mobile network parameters.

TABLE I: Mobile network parameters

Variable	Description
G_{net}	Mobile network graph.
N_{net}	Set of nodes in G_{net}.
N_{core}	Set of cores in G_{net}.
N_{du}, N_{cu}, N_{ndu}	Set of, respectively, DUs, CUs, non-DUs in G_{net}.
N^c_{du}	Set of DUs connected to CU $c \in N_{cu}$ in G_{net}.
N_{vsf}	Set of virtualized service functions (VSFs).
N^s_{ins}	Set of instances of the VSF $s \in N_{vsf}$.
N_{cls}	Set of service classes.
E_{net}	Set of FH and BH links in G_{net}.
$\omega^s_{num}(n)$	UEs that can share VSF $s \in N_{vsf}$ on $n \in N_{net}$.
$\omega_{cpu}(n)$	Processing capacity of the node $n \in N_{net}$.
$\omega_{prb}(d)$	Number of PRBs of DU $d \in N_{du}$.
$\omega_{bwt}(e^{nm})$	Capacity of the link $e^{nm} \in E_{net}$.

TABLE II: SFC request parameters

Variable	Description
G_{req}	UE's SFC request graph.
N_{ue}	Set of UEs in G_{req}.
N^u_{sfc}	Set of VSFs in the SFC request of UE $u \in N_{ue}$.
E_{req}	Set of all virtual links in G_{req}.
$E_{req}(u)$	Set of virtual links of the UE $u \in N_{ue}$.
$\omega^u_{bwt}(e')$	Data rate demand of link $e' \in E_{req}(u)$ of UE $u \in N_{ue}$.
$\omega^u_{prb}(d)$	PRB demand of UE $u \in N_{ue}$ from DU $d \in N_{du}$.

C. Service Function Chain (SFC) Request Model

SFC requests are modelled as *directed* graphs $G_{req} = (N_{req}, E_{req})$ where $N_{req} = N_{ue} \cup N_{sfc}$ is the union of the

set of UEs and their requested SFCs, while E_{req} is the set of virtual links between UEs and their SFCs, and the links between VSFs that make up SFCs.

Each UE generates a certain amount of data per second to be processed by the requested SFC characterized by a maximum acceptable E2E latency T_{E2E} (e.g., strict, medium, loose) and data rate ω^u_{bwt}. T_{E2E} is computed from the time UEs start transmitting data in the uplink (UL) until the time they receive and process the data in the downlink (DL) as follows:

$$T_{E2E} = T^{air}_{tr} + T^{air}_{prp} + T^{du}_{prc} + T^{fh,bh}_{tr} + T^{fh,bh}_{prp} + T^{sfc}_{exc} + T^{ue}_{prc} \tag{1}$$

where $T^{air}_{tr}, T^{air}_{prp}$ and $T^{fh,bh}_{tr}, T^{fh,bh}_{prp}$ are transmission and propagation time, respectively, over the air and FH/BH links, and T^{du}_{prc} is the baseband processing time in both UL and DL directions. Lastly, T^{vsf}_{exc} is the VSF execution time, and T^{ue}_{prc} is the UE processing time in DL. Since in reasonable settings the target block error rate (BLER) in mobile networks is 10% [19], we mimic hybrid automatic repeat request (HARQ) re-transmissions by considering the data size to be transmitted and processed by the SFC 10% more the data generated by UEs. It is worthwhile to mention that, although in the considered scenario data is transmitted and received by the same UE, the system model can be easily adapted to consider also the case in which data may be transmitted by one UE in UL and after processing be received by an other UE in DL. The see-through use case in car-to-car communication [20] is a descriptive example of such a scenario.

Notice that VSFs can be served from any node as long as the network has sufficient resources and the E2E latency along with the data rate requirements are respected. Figure 1b illustrates samples of SFC requests, while Fig. 1c shows SFC placement options minimizing, respectively, the E2E latency, the service provisioning cost and the VSF migration frequency. Each UE $u \in N_{ue}$ is also associated with a geographic location $loc(u)$, as x, y coordinates. Table II summarizes the UE request parameters.

IV. Problem Formulation

A. ILP Formulation

Before formulating the ILP model, for each UE, we first need to find the set of DUs that provide coverage. Considering the locations $loc(u)$ of the UE $u \in N_{ue}$ along with the location $loc(d)$ and the coverage radius $\delta(d)$ of DUs $d \in N_{du}$, the set of candidate DUs $\Omega(u)$ for the UE u can be defined as follows:

$$\Omega(u) = \Big\{ d \in N_{du} | dist(loc(d), loc(u)) \leq \delta(d) \Big\} \tag{2}$$

Additionally, we need to know the network nodes (e.g., DUs, CUs, the core) that can host VSFs of the SFCs requested by UEs. For each UE $u \in N_{ue}$, the set of candidate nodes $\tilde{\Omega}(u)$ can be defined as follows:

$$\tilde{\Omega}(u) = \Big\{ d \in \Omega(u), c \in N_{cu}, \hat{c} \in N_{core} | e^{d,c}, e^{c,\hat{c}} \in E_{net} \Big\} \tag{3}$$

Thus, in our model, either the UE's candidate DU, or the CU connected to the candidate DU, or the core node connected to the CU hosting the candidate DU can serve the UE's SFC.

TABLE III: Binary decision variables $\{0, 1\}$

Variable	Description
ξ^u_d	Indicates UE $u \in N_{ue}$ association with DU $d \in N_{du}$.
$\Phi^{u,s}_{n,i}$	Indicates if the VSF $s \in N^u_{sfc}$ requested by the UE $u \in N_{ue}$ has been served by the $i \in N^v_{ins}$ of the node $n \in N_{net}$.
$\Phi^s_{n,i}$	Indicates if any UE uses the i_{th} instance of the VSF $s \in N_{vsf}$ of the node $n \in N_{net}$.
$\Phi^{u,\hat{u}}_{\tilde{n}}$	Indicates if any VSF of UE $\hat{u} \in N_{ue}$ has been served by the non DU $\tilde{n} \in N^u_{ndu}$ of UE $u \in N_{ue}$.
$\zeta^{u,e'}_e$	Indicates if the virtual link $e' \in E_{req}(u)$ of the UE $u \in N_{ue}$ has been mapped to the substrate link $e \in E_{net}$.

Table III shows all binary variable used in this ILP formulation. The first objective function (formula (4)) of this ILP formulation aims at minimizing the E2E latency to serve SFCs.

$$\begin{aligned} min \Bigg(& \sum_{u \in N_{ue}} \sum_{d \in N_{du}} \Big(T^{air}_{tr}(d) + T^{air}_{prp}(d) + T^{du}_{prc}(d) \Big) \xi^u_d + \\ & + \sum_{u \in N_{ue}} \sum_{n \in N_{net}} \sum_{s \in N_{vsf}} \sum_{i \in N^s_{ins}} T^s_{exc}(u) \Phi^{u,s}_{n,i} + \\ & + \sum_{u \in N_{ue}} \sum_{\hat{u} \in N_{ue}} \sum_{\tilde{n} \in N^u_{ndu}} T^{fh,bh}_{tr}(\hat{u}, \tilde{n}) \Phi^{u,\hat{u}}_{\tilde{n}} + \\ & + \sum_{u \in N_{ue}} \sum_{e \in E_{net}} \sum_{e' \in E_{req}} T^{fh,bh}_{prp}(e) \zeta^{u,e'}_e \Bigg) \end{aligned} \tag{4}$$

Note that formula (4) does not take into account T^{ue}_{prc} since its constant for a given UE and data size and is independent of any decision variable. It is worth to mention that since the VSF instances and the FH/BH links are shared in the mobile network, T^s_{exc} and $T^{fh,bh}_{tr}$ depend on the number of UEs using, respectively, the same VSF instance on the same DC and the same FH/BH link.

The second objective function (formula 5) aims at minimizing the overall SFC provisioning cost. This encompasses the PRB usage cost Λ_{prb} (per PRB), the cost for using FH/BH bandwidth resources Λ_{bwt} (per Mbps) and the CPU usage cost Λ_{cpu} (per CPU) with the latter being much more expensive than the former ones. While Λ_{prb} is the same for all DUs and Λ_{bwt} is the same for all links, the Λ_{cpu} depends on the node hosting the VSF. Specifically, the closer is the host node to DUs, the more expensive is the CPU usage cost to instantiate a VSF on that node. This cost selection approach is justified by the fact that the edge nodes posses less computing capacity compared to the core nodes.

$$\begin{aligned} min \Big(& \sum_{u \in N_{ue}} \sum_{d \in N_{du}} \Lambda_{prb} \omega^u_{prb}(d) \xi^u_d + \\ & + \sum_{u \in N_{ue}} \sum_{n \in N_{net}} \sum_{s \in N_{vsf}} \sum_{i \in N^s_{ins}} \Lambda_{cpu}(n) \Phi^{u,s}_{n,i} + \\ & + \sum_{u \in N_{ue}} \sum_{e \in E_{net}} \sum_{e' \in E_{req}} \Lambda_{bwt} \omega^u_{bwt}(e') \zeta^{u,e'}_e \Big) \end{aligned} \tag{5}$$

The last objective function (6) has the goal of minimizing the migration frequency of VSFs. As opposed to previous cases, in the case, the CPU usage cost $\Lambda_{cpu}(n, cl)$ depends not only on the node hosting the required VSF, but also on the service class

of the requested SFC. For example, if the UE requests a SFC that has a strict E2E latency requirement, it is cheaper to serve the SFC from a DU compared to CUs or the core. Conversely, if the SFC has loose E2E latency requirement, it is cheaper to serve the SFC at the core compared to CUs and DUs. This approach effectively leads to minimization of migrated VSFs since VSF migration, which mostly occurs in the previous mapping strategies, is triggered due to E2E service latency violation that stems from FH/BH and processing resource sharing.

$$min \sum_{u\in N_{ue}} \sum_{n\in N_{net}} \sum_{cl\in N^u_{cls}} \sum_{s\in N_{vsf}} \sum_{i\in N^s_{ins}} \Lambda_{cpu}(n,cl)\Phi^{u,s}_{n,i} \quad (6)$$

We will now detail the constraints used in this ILP formulation. Regardless of the objective function, all the following constraints have to be satisfied in order for a solution to be valid. Constraint (7) ensures that each UE is associated to one DU that belongs to its candidate set (8).

$$\sum_{d\in N_{du}} \xi^u_d = 1 \quad \forall u \in N_{ue} \quad (7)$$

$$\sum_{d\in N_{du}\setminus\Omega(u)} \xi^u_d = 0 \quad \forall u \in N_{ue} \quad (8)$$

Each VSF $s \in N^u_{sfc}$ of the SFC requested by the UE $u \in N_{ue}$ must be served only once (9) by either the UE's host DU, or the CU connected to the host DU or by the core node connected to the CU of the host DU (10).

$$\sum_{n\in\tilde{\Omega}(u)} \sum_{i\in N^s_{ins}} \Phi^{u,s}_{n,i} = 1 \quad \forall u \in N_{ue}, \quad \forall s \in N^u_{sfc} \quad (9)$$

$$\xi^u_d - \sum_{n\in\tilde{\Omega}(u,d)} \sum_{i\in N^s_{ins}} \Phi^{u,s}_{n,i} \le 0 \quad \forall u \in N_{ue}, d \in \Omega(u), s \in N^u_{sfc} \quad (10)$$

Constraint (11) enforces for each virtual link there will be a continuous path established between the DU hosting the UE and the node(s) serving the SFC. $E^{\star i}_{net}$ is the set of the links that originate from any node and directly arrive at the node $i \in N_{net}$, while $E^{i\star}_{net}$ is the set of links that originates from the node i and arrive at any node directly connected to i.

$$\sum_{e\in E^{\star i}_{net}} \zeta^{e^{n,m}}_e - \sum_{e\in E^{i\star}_{net}} \zeta^{e^{n,m}}_e = \begin{cases} -1 & \text{if } i = n \\ 1 & \text{if } i = m \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

$$\forall i \in N_{net}, \quad \forall e^{n,m} \in E_{req}$$

Virtual links can be mapped to a FH/BH link in the mobile network as long as the link has enough capacity to meet the data rate demand of the virtual links (12).

$$\sum_{u\in N_{ue}} \sum_{e'\in E_{req}(u)} \omega^u_{bwt}(e')\zeta^{u,e'}_e \le \omega_{bwt}(e) \quad \forall e \in E_{net} \quad (12)$$

While constraint (13) makes sure that the computing capacity of the nodes is not exceeded, where $\Phi^s_{n,i} = 1$ if $\sum_{u\in N_{ue}} \Phi^{u,s}_{n,i} \ge 1$, constraint (14) sets an upper limit on the number of UEs that can share the same VSF.

$$\sum_{s\in N_{vsf}} \sum_{i\in N^s_{ins}} \Phi^s_{n,i} \le \omega_{cpu}(n) \quad \forall n \in N_{net} \quad (13)$$

$$\sum_{u\in N_{ue}} \Phi^{u,s}_{n,i} \le \omega^s_{num}(n) \; \forall n \in N_{net}, s \in N_{vsf}, i \in N^s_{ins} \quad (14)$$

The following constraint guarantees that the E2E latency to serve the UE $u \in N_{ue}$ does not violate the latency limit of the service requested by the UE.

$$\sum_{d\in N_{du}} \big(T^{air}_{tr}(d) + T^{air}_{prp}(d) + T^{du}_{exc}(d)\big)\xi^u_d + T^{ue}_{prc}(u) + \sum_{\hat{u}\in N_{ue}} \sum_{\tilde{n}\in N^u_{ndu}} T^{fh,bh}_{tr}(\hat{u},\tilde{n})\Phi^{u,\hat{u}}_{\tilde{n}} + \sum_{e\in E_{net}} \sum_{e'\in E_{req}} T^{fh,bh}_{prp}(e)\zeta^{u,e'}_e \sum_{n\in\tilde{\Omega}(u)} \sum_{\hat{u}\in N_{ue}} \sum_{s\in N^u_{sfc}} \sum_{i\in N^s_{ins}} T^s_{exc}(u)\Phi^{\hat{u},s}_{n,i} \le T_{lim}(u) \; \forall u \in N_{ue} \quad (15)$$

Finally, since the FH/BH links are shared among the UEs that use those links, the transmission time over the links $T^{fh,bh}_{tr}$ depends on the aggregated data size to be transmitted over the links. Constraint (16) handles accurate $T^{fh,bh}_{tr}$ calculation for each UE, considering three cases. Specifically, if the VSF of the UE has been mapped on the core node ($\tilde{n} \in N_{core}$) then for each CU ($\forall c \in N_{cu}$), it is checked if there are other UEs that have been associated to the same DU or to different DUs being connected to the same CU. If such UEs exist then it is checked if VSFs of those UEs are served by the core nodes ($C1$) or they are served by the CU connected to the host DU ($C2$). Whereas, $C3$ handles the case in which the UE's VSF has been served by a CU and there are other UEs who have been associated with the same host DU $\tilde{d} \in N^n_{du}$ with their VSFs being served either by the same CU or by the core $\hat{c} \in N_{core}$ connected to the CU linked to the first UE's host DU. After checking all possible VSF mappings, $T^{fh,bh}_{tr}$ is calculated for all UEs taking into account other UEs' data that use the same links. For instance, the red UE's data should be considered in $T^{fh,bh}_{tr}$ calculation of the blue UE (see the middle SFC placement example in Fig. 1c) over both FH and BH links since they both are used by the UEs. If the red UE's SFC was served by the CU then the red UE would affect $T^{fh,bh}_{tr}$ of the blue UE only over the FH link since only the FH would be used by both UEs.

$$\begin{cases} \sum_{d\in N^c_{du}} (\xi^u_d + \xi^{\hat{u}}_d) + \Phi^{u,s}_{\tilde{n},i} + \Phi^{\hat{u},\hat{s}}_{\tilde{n},i} - \Phi^{u,\hat{u}}_{\tilde{n}} \le 3 & C1 \\ \sum_{d\in N^c_{du}} (\xi^u_d + \xi^{\hat{u}}_d) + \Phi^{u,s}_{\tilde{n},i} + \Phi^{\hat{u},\hat{s}}_{c,i} - \Phi^{u,\hat{u}}_{\tilde{n}} \le 3 & C2 \\ \xi^u_d + \xi^{\hat{u}}_d + \Phi^{u,s}_{\tilde{n},i} + \Phi^{\hat{u},\hat{s}}_{\tilde{n},i} - \Phi^{u,\hat{u}}_{\hat{c}} \le 3 & C3 \end{cases} \quad (16)$$

$$\forall u \in N_{ue}, \tilde{n} \in N^u_{ndu}, s \in N^u_{sfc}, i \in N^s_{ins}, \hat{u} \in N_{ue}\setminus u, \hat{s} \in N^{\hat{u}}_{sfc}$$

B. Heuristic

The ILP formulation becomes computationally intractable as the size of the mobile network increases, e.g., the number of DUs/CUs, the variety of VSFs, the complexity of SFCs. For example, the ILP algorithm takes a day on Intel Core i7 laptop (3.0 GHz CPU, 16 Gb RAM) using the ILOG CPLEX

Algorithm 1: *Heuristic*

Data: (G_{net}, G_{req})
Result: UEs association, SFC placement and resource allocation.
Step 1: Find candidate nodes per UE and VSF demand per SFC class per DU;
for $d \in N_{du}$ **do**
  for $cl \in N_{cls}$ **do**
    $sfc\_cls(d, cl) \leftarrow \emptyset$;
for $u \in N_{ue}$ **do**
  $cand\_du(u), cand\_vsf(u) \leftarrow \emptyset$;
  for $d \in N_{du}$ **do**
    $dist \leftarrow dis(loc(u), loc(d))$;
    if $dist \leq \delta(d)$ **then**
      $cand\_du(u), cand\_vsf(u) \leftarrow d$;
      $cand\_vsf(u) \leftarrow N^d_{cu}$;
      for $s \in N^u_{sfc}$ **do**
        $sfc\_cls(d, N^u_{cls}, s) \leftarrow sfc\_cls(d, N^u_{cls}, s) + 1$;
  $cand\_vsf(u) \leftarrow N^d_{core}$;
Step 2: Find VSF mapping per node;
for $n \in N_{net}$ **do**
  $map\_cand\_vsf(n) \leftarrow \emptyset$;
for $d \in N_{du}$ **do**
  • *Loose class delay VSFs mapping* $N^d_{core} -> N^d_{cu} -> d$;
  • *Medium class delay VSFs mapping* $N^d_{cu} -> N^d_{core} -> d$;
  • *Strict class delay VSFs mapping* $d -> N^d_{cu} -> N^d_{core}$;
  • *Populate* map_cand_vsf *per VSF host*;
Step 3: Perform UE association;
for $u \in N_{ue}$ **do**
  $m_c(u) \leftarrow 0$;
  for $p \in cand\_du(u)$ **do**
    for $s \in N^u_{sfc}$ **do**
      $c_{curr} \leftarrow +\infty$;
      for $q \in cand\_vsf(u)$ **do**
        if $s \in map\_cand\_vsf(q)$ **then**
          $c_{new} \leftarrow c_{link}(p, q) + c_{node}(p)$;
          $c_{curr} \leftarrow min(c_{curr}, c_{new})$;
      $m_c(p) \leftarrow m_c(p) + c_{curr}$;
  $p \leftarrow argmin(m_c(p))$;
  $mapped(u) \leftarrow p$;
  Step 4: Perform SFC placement and resource allocation;
  for $s \in N^u_{sfc}$ **do**
    $m_c(s) \leftarrow 0$;
    $c_{curr} \leftarrow +\infty$;
    for $q \in cand\_vsf(u)$ **do**
      • *Compute* $T_{E2E}(u)$;
      if $s \in map\_cand\_vsf(q)$ **then**
        for $i \in inst\_vsf(s)$ **do**
          if $map\_cand\_vsf(p)\{s, i\} \leq 0$ ***or*** $T_{E2E}(u) > T_{lim}(u)$ **then**
            continue;
          $c_{new} \leftarrow c_{link}(p, q) + c_{node}(p)$;
          if *no* T_{lim} *violatation for any UE* **then**
            $c_{curr} \leftarrow min(c_{curr}, c_{new})$;
          $m_c(q) \leftarrow c_{curr}$;
    $q \leftarrow argmin(m_c(s))$;
    $mapped(s) \leftarrow q$;
    • *Allocate path* $P_{p,q}$;
    • *Allocate and update network resources*;
    • *Recompute* T_{lim} *for all UEs* ;

12.8 solver to associate and serve 300 UEs making latency-sensitive SFC requests each composed of three VSFs in the network composed of 4 DUs, 2 CUs and a core. In order to address this scalability issue, we develop a heuristic, as shown in Algorithm 1, that is able to embed the same requests in less than a second.

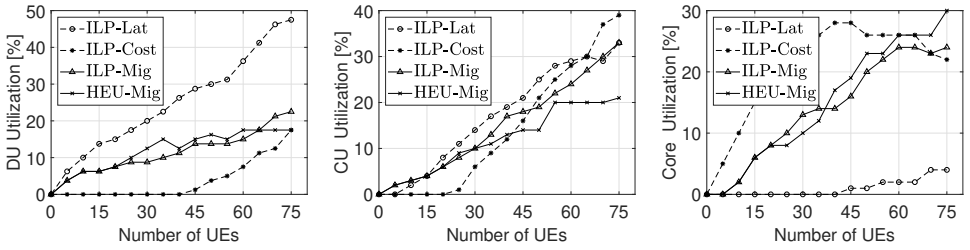
The proposed heuristic has an objective of minimizing the number of VSF instance migration, which is achieved in four steps. In the first step, the heuristic initiates sfc_cls vector to keep the count of each VSF demand per service class per DU. Then, the heuristic creates a list of candidate DUs $cand\_du$ for each UE by looping over all DUs, considering the coverage radius of each DU and the distance between the DU and the UE. Additionally, the heuristic creates a list of candidate nodes $cand\_vsf$ for VSFs in the SFC requested by the UE.

In the second step, the algorithm considers all VSFs on each DU, and the VSF instantiation starts from the VSFs that belong to the SFCs with the loose latency class towards the ones with the strict latency class. Specifically, for each VSF from the loose service latency class, the algorithm first checks that if a VSF is already available on the core. If it is not available or does not have enough capacity to support the UEs' demand, it instantiates a new VSF on the core. This process is repeated on the CU connected to the DU and then on the DU itself until the VSF is instantiated on one of these nodes. Once it has been instantiated, the sfc_cls vector is updated subtracting those UEs' VSF demand that are under the coverage of the DU that hosted the VSF or is connected to the node hosting the VSF. A similar process is performed for the medium latency class and the strict latency class VSFs with the order of, respectively, CU, core, DU and DU, CU, core. In the end, sfc_cls becomes a vector of zeros for all latency classes, indicating that all VSFs of the requested SFCs have been instantiated, and map_cand_vsf matrix is derived containing VSF instances on all nodes.

In the third step, the algorithm performs UEs' association in the following manner. For each UE, the algorithm traverses all its candidate DUs for each considering every VSF of the SFC requested by the UE and computing its placement cost on its those candidates that already have the VSF instance. A VSF placement cost encompasses both the link and the node resource usage costs. At the end of this step, the heuristic picks the DU for the UE association that would result in the minimal UE association and its SFC placement cost. Finally, in the last step, the heuristic places the SFC requested by the UE and allocates required resources. Specifically, for each VSF of the UE's SFC, the heuristic computes the E2E latency on each VSF instance of each candidate node that has the requested VSF. This is followed by checking if the VSF placement on the candidate node violates the latency class limit of the UE. If the VSF placement does not violate any UE's latency limit then the algorithm will compute the mapping cost. After repeating this process for all the VSF candidate DCs, the algorithm will map the VSF to the DC that would serve the VSF with the minimal cost. Lastly, the network resources will be allocated and T_{lim} time limit will be re–estimated for all the UE.

V. Evaluation

The goal of this section is to compare the ILP–based solutions, *ILP-Lat, ILP-Cost* and *ILP-Mig*, which aim at minimizing E2E SFCs' latency, service provisioning cost, and VSF migration frequency, respectively, and *Heu-Mig*, the heuristic counterpart of *ILP-Mig*. We shall first describe the simulation setup. We will then discuss the outcomes of the numerical simulations carried out in a simulator implemented in Matlab.



(a) CPU utilization of DUs. (b) CPU utilization of CUs. (c) CPU utilization of core.

Fig. 2: CPU utilization of DU, CU and core nodes.

A. Simulation Environment

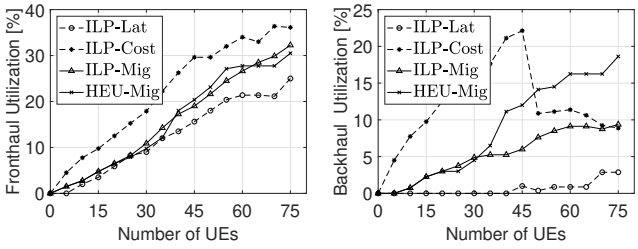
The mobile network considered in the simulations is composed of 7 nodes, similar to the one depicted in Figure 1a. The core node is connected to the CUs by means of 20Gbps fiber BH links, while the CUs are connected to the DUs by means of 10Gbps wireless FH links. The core node, CUs and DUs have, respectively, 10, 6 and 2 CPUs, and it is assumed that each VSF requires a single CPU in order to be spawned/instantiated. Once a VSF has been instantiated at a network node that VSF can be shared among maximum of 10 UEs as long as the E2E latency requirement imposed by the services requested by the UEs is not violated due to the aggregated task execution time of the VSF. In the simulations, SFC requests arrive in batches each of which corresponding to a timeslot composed of 5 UEs making SFC requests. With each batch, the algorithms try to associate all UEs (also the ones from the previous timeslots) and serve their SFC requests. Due to scalability issue of the ILP–based algorithm, we consider 15 batches of SFC requests (75 UEs). Each SFC consists of VSFs, whose quantity is randomly picked from the set of $\{2, 3, 4\}$, which are then randomly picked from 10 VSFs. VSFs in an SFC are sequentially connected to each other, similar to the one depicted in Figure 1b. Depending on the service class, the network provider has to guarantee a certain E2E latency and data rate requirements. Specifically, we consider three service classes having, respectively, $[20, 50, 100]\mu$s E2E latency, $[400, 200, 150]$Mbps data rate requirements and generating $[1, 5, 9]$Mbit data/task per second to be processed by the requested SFC.

Note that like in [21], we assume that sufficient PRBs are always allocated to the UEs in order to keep high QoS and make sure that the data rate requirement of their requested SFC is always satisfied. Moreover, since the focus of this work is mostly on the SFC placement problem, the selection of a particular UE channel model, although important, takes a secondary role. As a result, in the numerical evaluation, we leverage on a simple modulation and coding scheme (MCS) estimation model which is based on the distance between the UE and the host DU. Finally, for the sake of simplicity, it is assumed that the data size and data rate both in DL and UL remain the same. While T_{tr}^{air} is computed for each individual UE by dividing the data generated by UE by its data rate, $T_{tr}^{fh,bh}$ and T_{exc}^{sf} are computed for all UEs employing, respectively, the same FH/BH link and VSF since FH/BH links and VSFs are shared resources. Specifically, $T_{tr}^{fh,bh}$ for the UEs using the same FH/BH link at the considered moment is obtained by dividing the aggregated data size by the FH/BH link rate. Thus, $T_{tr}^{fh,bh}$ is the same for all the UEs using the same FH/BH link. Whereas, T_{exc}^{sf} is the ratio between the product of the aggregated data size to be processed by the VSF and the number of CPU cycles for processing a single bit of data, and the CPU capacity. T_{prc}^{ue} is computed in a similar fashion for each UE. A single CPU capacity of a network node and a UE is, respectively, 3.5 GHz and 1.5 GHz. Lastly, baseband processing time T_{prc}^{du} at DUs is computed according to [22].

B. Simulation Results

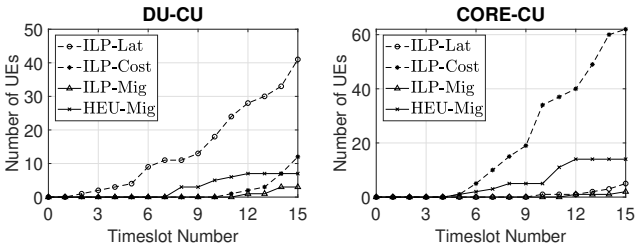
CPU utilization. Since VSFs can be shared among several UEs, and a single VSF requires one CPU to be instantiated, CPU capacity of a node is expressed in terms of the number of UEs that can employ VSFs/CPUs on that node and is equal to the number of CPUs available at the node times the number of UEs that can use the same VSF/CPU. Consequently, CPU utilization of a node is computed by dividing the number of UEs using VSFs of that node by the overall capacity of the node. Let us now analyze the CPU utilization of DU, CU and core DCs for presented algorithms (Fig. 2) in a single simulation run. In Fig. 2a, we can observe that the CPU utilization at the DUs is the highest for the *ILP-Lat* algorithm due to the fact that, regardless of the E2E latency requirement of the requested service, *ILP-Lat* algorithm tends to instantiate VSFs at the DUs as long as they have enough CPUs. Conversely, CPU utilization at the DUs is the lowest for the *ILP-Cost*. Indeed, we can observe that *ILP-Cost* starts placing VSFs at the DUs when the number of UEs in the network is 45. This can be justified by the fact that up to 40 UEs, *ILP-Cost* provisions VSFs from the CUs and the core. However, when the VSF demand increases, some VSFs are placed at the DU in order to meet UEs' E2E latency requirements. As for *ILP-Mig* and *Heu-Mig* algorithms, they pick the nodes for instantiating VSFs by considering the latency class of the requested SFC, ultimately achieving similar CPU utilization that lies between the ones achieved by *ILP-Lat* and *ILP-Cost*. Thus, they do not initially tend to consume the computational resources of only DUs or the core, like *ILP-Lat* and *ILP-Lat*, respectively, neglecting the latency class of the requested SFC.

Figure 2b displays the CPU utilization at the CUs for all algorithms. It can be seen that the gap between CPU utilization achieved by the algorithms is narrow. This stems from the fact that apart from *ILP-Mig* and *HEU-Mig*, *ILP-Lat* and *ILP-Cost* start serving VSFs from CUs. Specifically, *ILP-Lat*



(a) FH utilization. (b) BH utilization.

Fig. 3: FH and BH link utilization.



(a) UEs whose VSF provisioning is changed from DU to CU.

(b) UEs whose VSF provisioning is changed from core to CU.

Fig. 4: UEs whose VSF provisioning has been changed.

starts placing VSFs at CUs because of the lack of CPU resource at the DUs, while *ILP-Cost*, as some point, starts placing VSFs at CUs in order to satisfy E2E latency demand of the UEs. As for CPU utilization at the core (see Fig. 2c), we can observe a reverse trend compared to the one at DUs. Specifically, we can observe that no VSF is spawned at the core by *ILP-Lat* algorithm up to 40 UEs. Whereas, when the number of UEs demanding SFCs increases, VSFs start being served by the core due to the saturation of CPU resources at the DUs and CUs. As opposed to *ILP-Lat*, *ILP-Cost* increases the CPU utilization at the core up to 40 UEs, while with further increase in the service demand, the CPU utilization plummets as a result of migrating some VSFs from the core to CUs. Like in Fig. 2a, we can observe that the CPU utilization by *ILP-Mig* and *HEU-Mig* algorithms resembles being in between the ones achieved by *ILP-Lat* and *ILP-Cost*.

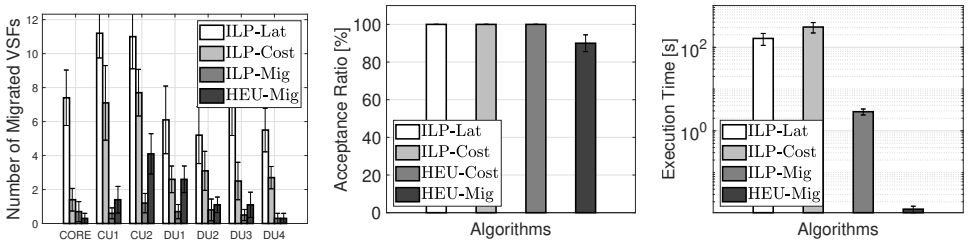
Link utilization. Figure 3 illustrates the FH and BH link utilization as a function of the number UEs for the same single simulation run. We can observe that *ILP-Cost* algorithm achieves the highest level of FH and BH utilization up to 45 UEs making SFC requests. This is due to the fact that *ILP-Cost* strives to place VSFs at the core as long as the E2E latency requirement of the services requested by UEs is not violated. We can also observe that while with more UEs the FH utilization exhibits an increasing trend, the BH utilization drops significantly as a result of VSF migration from the core to CUs and DUs. As for *ILP-Lat* algorithm, it achieves the lowest FH and BH utilization due to its optimization objective. It is interesting to notice in Fig. 3b that up to 40 UEs, the requested VSFs are provisioned from DUs and CUs since no BH link is used. Similar to CPU utilization plot, FH utilization for all UEs and BH utilization up to 40 UEs in the cases of *ILP-Mig* and *HEU-Mig* algorithms lies between the ones achieved by *ILP-Lat* and *ILP-Cost* algorithms. Thus compared to *ILP-Lat* and *ILP-Cost* algorithms, the ILP–based and heuristic migration algorithms find better compromise between CPU utilization and FH/BH utilization.

VSF provisioning DC change. Figure 4 shows the accumulated number of UEs whose VSF provisioning DC has been changed from DUs to CUs (see Fig. 4a) and from core to CUs (see Fig. 4b)[1] with the arrival of UEs for the same single iteration. Since *ILP-Lat* seeks to place VSFs as close to DUs as possible, it results in the highest number of UEs changing their VSF provisioning from DUs to CUs and only a few UEs changing from the core to CUs. This is due to limited CPU capacity of DUs. Conversely, the objective of *ILP-Cost* algorithm causes the highest number of UEs to change their VSF provisioning from the core to CUs and only a few UEs changing their VSF provisioning from DUs to CUs. As for *ILP-Mig* and *HEU-Mig* algorithms, we can observe that the overall number of UEs that change their VSF provisioning is less compared to the ones achieved by *ILP-Lat* and *ILP-Cost* algorithms. We can also observe that *ILP-Mig* results in the lowest number of UEs' VSF provisioning change due the optimality of found solutions as opposed to the ones found by *HEU-Mig*.

Quantity of migrated VSFs per DC. In order to get an insight into how migration of VSFs takes place between different network nodes, let us analyze Fig. 5a, which illustrates the average number of migrated VSFs at each node/DC for 10 simulation runs. As expected, the highest number of VSF migrations take place when *ILP-Lat* algorithm is used. This stems from the fact *ILP-Lat* starts instantiating VSFs from DUs towards the core, and since the CPU capacity of the nodes is limited, *ILP-Lat* makes VSF placement decisions based on their demand, resulting in the highest number of VSF migration. The second highest number of VSF migrations are caused by *ILP-Cost* algorithm. The main reason for this is that migration of VSFs is triggered due to E2E latency requirement of the requested services since *ILP-Cost* starts placing VSFs from the core towards DUs, entailing to high transmission delay over FH/BH links, which may result in a rejection of UEs SFC requests unless VSFs are migrated from the core towards the edge. As for *ILP-Mig* and *Heu-Mig* algorithms, due to their objective function (see formulas (4) and (5)), they migrate fewer VSFs from each node compared to the rest of the algorithms. Among the algorithms minimizing the number of VSF migrations, *ILP-Mig* achieves a fewer VSF migration since, as opposed to *Heu-Mig*, it is always able to find an optimal VSF placement solution.

Acceptance ratio. Since all constraints defined in Section IV-A for the ILP–based algorithms are imposed on all of them, although with different VSF placements due to different objective functions, they accept an equal number of UEs. Specifically, Fig. 5b shows that the three ILP–based algorithms accept all SFC requests of UEs during all simulations. Whereas, due to suboptimal VSF placements, *Heu-Mig* accepts 90% of UEs' SFC requests on average with the maximum of 4.5% difference from the mean values in their confidence intervals.

[1]Note that the plots showing CU–DU and CU–CORE VSF provisioning change are omitted due to space limitation.



(a) VSF migration per node. (b) Acceptance ratio. (c) Execution time.

Fig. 5: Quantity of migrated VSFs per DC, acceptance ratio and execution time for all algorithms.

Execution time. The main motivation for proposing the heuristic is to address the scalability issue of the ILP–based algorithms. Fig. 5b shows the average execution time of associating a single SFC request for all algorithms. Among the ILP–based algorithms, it can be observed that the execution time for *ILP-Mig* is around 100 times lower than those of *ILP-Lat* and *ILP-Cost*. The rationale behind this is the simplicity of the objective function of *ILP-Mig* (see formula 6) in comparison with the ones of *ILP-Lat* and *ILP-Cost* (see formulas (4) and (5)). Nevertheless, the execution time of *ILP-Mig* is significantly more (around 200 times) compared to the *Heu-Mig* algorithm. Thus, the heuristic algorithm demonstrates a trade–off between the optimality (e.g., acceptance ratio) and the scalability of the solution.

VI. Conclusions

Endowing edge nodes in 5G networks with computing capabilities is perceived to be a promising approach to curtail the E2E service latency, and therefore, be able to support novel services with their stringent QoS requirements. However, since the computing capacity of edge nodes is limited, not all services can reap the benefits of mobile edge computing.

In this study, we compared three strategies for solving a joint UE association, SFC placement and resource allocation problem. We have seen that while *ILP-Lat* improved QoS of all UEs by placing their SFCs close to DUs, saving FH/BH link resources, it has lead to a high number of VSF migrations, changing the SFC placements for many UEs from DUs to CUs. Moreover, it has underutilized computing resources in the core DC. Conversely, although *ILP-Cost* has better utilized the computing resources in the core DC, resulting in a reduced service provisioning cost for the network provider, it has significantly increased the FH/BH link utilization. Additionally, it has entailed many VFS migrations, changing the SFC placements for many UEs from the core to CUs. Whereas, *ILP-Mig* and *Heu-Mig* have eliminated these downsides finding a better compromise between the FH/BH link utilization and the computing resource utilization of the DCs while achieving less VSF migration and SFC placement change. Among these algorithms, *Heu-Mig*, at the expense of suboptimal UE associations and SFC placements, has demonstrated the fastest execution time, making it suitable for larger-scale problems.

References

[1] G. P. A. W. Group, "View on 5g architecture," *White Paper, July*, 2016.

[2] 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.

[3] I. Parvez, A. Rahmati, I. Guvenc, A. I. Sarwat, and H. Dai, "A survey on low latency towards 5G: RAN, core network and caching solutions," *IEEE Communications Surveys & Tutorials*, 2018.

[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] K. Katsalis, N. Nikaein, E. Schiller, R. Favraud, and T. I. Braun, "5g architectural design patterns," in *Proc. of IEEE ICC*, Kuala Lumpur, Malaysia, 2016.

[6] L. Tong, Y. Li, and W. Gao, "A hierarchical edge cloud architecture for mobile computing," in *Proc. of IEEE INFOCOM*, San Franc, USA, 2016.

[7] T. Zhao, S. Zhou, X. Guo, Y. Zhao, and Z. Niu, "A cooperative scheduling scheme of local cloud and internet cloud for delay-aware mobile cloud computing," in *Proc. of IEEE GLOBECOM Workshops*, San Diego, USA, 2015.

[8] H. Chang, H. Liu, Y.-W. Leung, and X. Chu, "Minimum latency server selection for heterogeneous cloud services," in *Proc. of IEEE GLOBECOM*, Austin, USA, 2014.

[9] A. Alleg, T. Ahmed, M. Mosbah, R. Riggio, and R. Boutaba, "Delay-aware vnf placement and chaining based on a flexible resource allocation approach," in *Proc. of IEEE CNSM*, Tokyo, Japan, 2017.

[10] S. Agarwal, F. Malandrino, C.-F. Chiasserini, and S. De, "Joint VNF Placement and CPU Allocation in 5G," in *Proc. of IEEE INFOCOM*, Hawaii, USA, 2018.

[11] M. Huang, W. Liang, Y. Ma, and S. Guo, "Throughput maximization of delay-sensitive request admissions via virtualized network function placements and migrations," in *Proc. of IEEE ICC*, Kansas, USA, 2018.

[12] H. Hawilo, M. Jammal, and A. Shami, "Orchestrating network function virtualization platform: Migration or re-instantiation?" in *Proc. of IEEE CloudNet*, Prague, Czech Republic, 2017.

[13] B. Martini, F. Paganelli, P. Cappanera, S. Turchi, and P. Castoldi, "Latency-aware composition of virtual functions in 5g," in *Proc. of IEEE NetSoft*, London, U.K, 2015.

[14] L. Qu, C. Assi, and K. Shaban, "Delay-aware scheduling and resource optimization with network function virtualization," *IEEE Transactions on Communications*, vol. 64, no. 9, pp. 3746–3758, 2016.

[15] G. Chochlidakis and V. Friderikos, "Low latency virtual network embedding for mobile networks," in *Proc. of IEEE ICC*, Kuala Lumpu, Malaysia, 2016.

[16] M. T. Beck and C. Linnhoff-Popien, "On delay-aware embedding of virtual networks," in *Proc. of AFIN*. Citeseer, 2014.

[17] K. Ivaturi and T. Wolf, "Mapping of delay-sensitive virtual networks," in *Proc. of IEEE ICNC*, Hawaii, USA, 2014.

[18] "5G; NG-RAN; Architecture description ," 3GPP TS 38.401 version 15.3.0 Release 15, Tech. Rep., 2018.

[19] " LTE; Feasibility study for Further Advancements for E-UTRA (LTE-Advanced)," 3GPP, Sophia Antipolis, France, 3GPP TR 36.912 version 14.0.0 Release 14, 2017.

[20] H. M. D. Sabella, "Toward fully connected vehicles: Edge computing for advanced automotive communications," White Paper, 2017.

[21] D. Harutyunyan, A. Bradai, and R. Riggio, "Trade-offs in cache-enabled mobile networks," in *Proc. of IEEE CNSM*, Rome, Italy, 2018.

[22] T. X. Tran, A. Younis, and D. Pompili, "Understanding the computational requirements of virtualized baseband units using a programmable cloud radio access network testbed," in *Proc. of IEEE ICAC*, Columbus, USA, 2017.