This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.3028216, IEEE Transactions on Mobile Computing

Latency and Mobility–Aware Service Function Chain Placement in 5G Networks

Davit Harutyunyan*, Nashid Shahriar‡, Raouf Boutaba§ and Roberto Riggio*†
*Smart Networks and Services, FBK, Italy; Email: d.harutyunyan,rriggio@fbk.eu
‡Department of Computer Science, University of Regina; Email: Nashid.Shahriar@uregina.ca
†i2CAT, Barcelona, Spain; Email: roberto.riggio@i2cat.net
§David R. Cheriton School of Computer Science, University of Waterloo, Canada;
Email: rboutaba@uwaterloo.ca

Abstract—5G networks are expected to support numerous novel services and applications 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 reduced by moving the computational capability closer to the network edge. The limited amount of computational resources of the edge nodes, however, poses the challenge of efficiently utilizing these resources while, at the same time, satisfying QoS requirements. In this work, we employ mixed-integer linear programming (MILP) techniques to formulate and solve a joint user association, service function chain (SFC) placement, where SFCs are composed of virtualized service functions (VSFs), and resource allocation problem in 5G networks composed of decentralized units (DUs), centralized units (CUs), and a core network (5GC). Specifically, we compare four approaches to solving the problem. The first two approaches minimize, respectively, the E2E latency experienced by users and the service provisioning cost. The other two instead aim at minimizing VSF migrations along with their impact on users' quality of experience with the last one minimizing also the number of inter-CU handovers. We then propose a heuristic to address the scalability issue of the MILP-based solutions. Simulations results demonstrate the effectiveness of the proposed heuristic algorithm.

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

1 Introduction

The 5th generation of mobile communication networks (5G) is on the horizon with the promise to revolutionize the communication landscape. 5G will enable a wide variety of services, including massive broadband, virtual/augmented reality (AR/VR), 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, reliability, and mobility [2], [3]. For instance, ultra-low latency services require 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 loose latency requirements.

To support the versatile and ambitious QoS requirements of different 5G services, the mobile network infrastructure is undergoing a paradigm shift towards adding distributed micro/edge data centers (DCs) [1]. This is enabled by the multi-access edge computing (MEC) technology, which brings the content and the computing resources closer to the end-users, making up a light data center (DC) and, therefore, curtailing the round-trip service provisioning latency and alleviating the transport network utilization. MEC servers can be collocated both with decentralized units (DUs) and centralized units (CUs) [4], which together make up a base station, called gNB in 5G terms, and can be statically deployed with traditional base stations. The DUs are equipped with antennas to serve the user equipments (UEs) and are connected to the CUs via fronthaul (FH) links, while the CUs, that are connected to the 5G core (5GC) via backhaul (BH) links, can serve multiple DUs. The 5GC will still be there providing an abundance of computing resources. Thus, the mobile network will be composed of distributed DCs (i.e, DU , CU, and 5GC DCs), where the closer is the DC to the 5GC, the more are its computing resources and, therefore, the cheaper it is to use these resources. For example, sub-millisecond latency services facilitating AR/VR may be composed of multiple service functions (SFs) some of which (e.g., video rendering) may need to be processed right at the DU DCs, thus avoiding the round-trip delay to and from either CU or 5GC DCs.

Another technology expected to play a pivotal role in 5G is virtualization that decouples SFs from dedicated proprietary hardware and deploys virtualized service functions (VSFs) on commodity servers, thus reducing CapEx [5], [6]. Virtualization provides the opportunity to deploy VSFs at DU, CU, and 5GC DCs, 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 shared among the VSFs belonging to different SFCs. However, sharing a VSF among multiple SFCs may increase both the 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 an 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 5GC DCs and FH/BH links, respectively. Therefore, instantiating an optimal number of VSFs in different DCs 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 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 [7]. 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 5GC DCs in order to accommodate newly arrived SFCs with strict latency requirements. Similarly, another strategy that initially places VSFs of SFCs in 5GC DCs irrespective of QoS requirements needs to adjust VSF placement later on. For instance, a VSF placed in a 5GC DC could satisfy a strict latency requirement when there is a light load and starts to violate its latency constraint as the load rises due to an 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 [8], [9]. In addition, when migration is the only viable option to satisfy UE's quality of experience (QoE) requirements, the SFC placement strategy should intelligently select a VSF to migrate so as to reduce the impact of the migration on the QoE of UEs.

Another factor that affects the joint UE association, SFC placement, and resource allocation problem is the user mobility. Such mobility can trigger a change in the UE association involving handover operation. The handover operation can be expensive depending on whether it is performed between DUs that belong to the same CU (called as intra-CU) or between DUs locating under two different CUs (called as inter-CU). While the inter-CU handover involves the 5GC for performing a path switch procedure to implement the CU change, the intra-CU handover can be performed without involving the core thanks to the centralized placement of Packet Data Convergence Protocol (PDCP) processing at the CU as per the 5G NG-RAN architecture [4]. Note that the centralized placement of PDCP at the CU is not feasible when a UE's VSF is placed at the DU to satisfy strict latency constraints. In order to take advantage of intra-CU handover, a mobile UE has to be associated with a new DU that belongs to the same CU of the UE's previous DU. Such a change in UE association can trigger a migration of a VSF influencing the sharing level of the VSF and the corresponding latencies. However, the impact of the UE association change on the number of VSF migrations is magnified when the handover is an inter-CU operation. Hence, another goal of an SFC placement strategy is to keep the number of inter-CU handovers as low as possible.

In this paper, we demonstrate the pros and cons of the aforementioned four SFC placement strategies through empirical simulation of a 5G mobile network. To do so, we employ mixed-integer linear programming (MILP) 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 mobile UEs. We also develop a comprehensive E2E latency model suitable for SFCs in the 5G mobile networks. In order to address the scalability issue of the MILPs, we then propose a heuristic that follows the objective of minimizing the number of inter-CU handovers, which exhibits the best performance.

This paper extends our initial work [10] in several aspects. First, we consider the mobility of UEs that further intricates the problem compared to the static UEs studied in [10]. Since user mobility induces a higher number of VSF migrations than in [10], we modify the objective of minimizing the number of VSF migrations to also include the impact of migrations on UEs' QoE. In addition, we introduce a new objective that minimizes the number of inter-CU handover operations triggered by user mobility apart from minimizing the number of VSF migrations and their impact. We also modify our heuristic algorithm to take into account the minimization of inter-CU handovers while keeping the impact of VSF migrations as low as possible. Finally, we analyse the performance of the compared approaches with additional metrics such as the number of intra-CU and inter-CU handovers and the utilization of physical resource blocks (PRBs), which are chunks of the time-frequency matrix in the radio access network and can be allocated to the UEs by the scheduler of the base station.

The rest of this paper is structured as follows. The related work is discussed in Sec. 2. The problem statement along with the mobile network and SFC request models are introduced in Sec. 3. The MILP problem formulation and the heuristic are presented in Sec. 4. The numerical results are reported in Sec. 5. Finally, Sec. 6 draws the conclusions.

2 Related Work

2.1 Server selection

One of the problems tackled in our study is the server selection in a heterogeneous cloud network for computation offloading. There is a sizable body of work published on this problem [11]–[13]. A hierarchical edge cloud architecture is proposed in [11] that offloads users' computational tasks to the clouds preferably closer to the users. The authors of [12] 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 strategy is presented in [13] that groups users 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 reducing the distance between clusters and servers. However, none of the aforementioned studies consider realistic latency-sensitive applications with E2E latency requirements envisioned to be supported in 5G networks.

2.2 SFC placement and scheduling

SFC placement and scheduling is a well-studied topic [14]–[20]. A number of works in this literature augments the SFC placement problem with a certain E2E latency requirement to be satisfied [21]–[23]. A delay-aware SFC placement problem is studied in [21] assuming that VSF processing delay linearly depends on the allocated resources. Nonetheless, this work considers the placement of VSFs on cloud servers neglecting the 5G mobile network architecture altogether. The authors in [22] address a VSF placement problem in a service-customized network where each SFC has a latency

bound based on the application. However, [22] uses a simplistic latency model where transmission delay is independent of the load, and the processing delay of VSFs is ignored.

Several other works, including [24]–[27], strive to minimize E2E latency while placing VSFs and scheduling SFCs. Among them, [25] studies the joint VSF placement and CPU allocation in 5G networks seeking to minimize the ratio between the actual and the maximum allowed latency across all services. Similarly, [26] addresses the joint optimization of SFC placement and request scheduling to minimize the average response latency. The authors in [28] adopt the idea of deploying a sequential SFC with parallel VSFs as long as there is no dependency on a packet imposed by these VSFs. However, these works do not consider heterogeneous servers located in hierarchical DCs, which augment the search space, making the SFC placement problem more cumbersome. Although [27] considers hierarchical DCs to perform resource allocation for ultra-low latency services, it ignores the delay in the air interface and user mobility.

Another thrust of relevant research formulates VSF scheduling and network resource allocation to establish an SFC while taking into account the latency of the SFC. For instance, the objective in [29] is to minimize the latency of the overall VSF's schedule to meet stringent delay requirements. On the other hand, [30] formulates the problem of composing and placing an SFC to nodes that minimizes the network and the processing latency. However, both of these works consider VSF instances already being placed in DCs and ignore the capacity constraints in the nodes. Similarly, [31] minimizes transport network latency to embed a set of VSFs for network slices, while ignoring VSF processing latency. Several other models, including [32]–[38], have been proposed for quantifying E2E latency in the context of a virtual network. Our proposed latency model stands out from these models in considering delays in the context of 5G networks including, delay in the transport network and VSF processing delays as a function of the users, respectively, sharing the transmission links and computational resources.

2.3 VSF instantiation and migration

The VSF instantiation and migration problem is studied in [39], 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 [40] employ an MILP model to decide whether to re-instantiate or migrate the VSFs and find their optimal placements while seeking to achieve minimal downtimes for the VSFs. In contrast, [41] studies the VSF migration problem with the goal of minimizing its effect on the overall network, which is defined as the sum of link delay difference before and after VSF migrations. Similarly, [9] strives to minimize the number of VSF migrations and to achieve load balancing in order to meet E2E delay requirements with time-varying traffic. However, [9] considers only processing delay of VSFs modeled using queueing theory ignoring transmission and propagation delay on links altogether. In contrast to existing studies on VSF migration, the focus of this paper is on minimizing the inter-CU handovers as well as the number of VSF migrations and their effect on the UEs' QoE for a 5G mobile network. In addition, the VSF migration decision in our study is influenced by several factors including mobility of users, E2E latency constraint satisfaction, and level of VSF sharing by different UEs as opposed to other studies.

2.4 Joint UE association and SFC placement

The work that considers user mobility and delay minimization while addressing access point selection and service placement for a number of users is [8]. This work discretizes the timeline of a MEC system into time slots and models the total queuing delay, communication delay, and the migration delay for all the users at each time frame. It strives to minimize the number of migrations by tolerating as much queuing delay and communication delay as possible until these delays exceed migration delay by a large margin. However, this work minimizes the total delays of the system where an individual user may experience a higher delay or its latency constraint may be violated. In contrast to all these related works, we consider the joint problem of SFC placement, user association, and network resource allocation that allows optimization of both computational and network resources based on users' location, services' demands and QoS requirements in terms of bandwidth and E2E latency.

3 Network Model

3.1 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 in which we assume that, like traditional evolved NodeBs (eNBs), DUs can still perform the entire baseband signal processing for those UEs whose VSFs are served from DUs, while if the UEs' requested VSFs are served by the CUs or the 5GC then the PDCP layer processing for all the DUs that UEs got associated with is centralized at their corresponding CUs [4]. Thus, each CU can serve multiple DUs over the FH links while each 5GC can serve multiple CUs over the BH links. In this hierarchical network architecture, each node (e.g., DU, CU, 5GC) can be thought of a DC that has a certain computational capacity, and it is assumed that the closer is the DC to the 5GC, the more is its computational capacity.

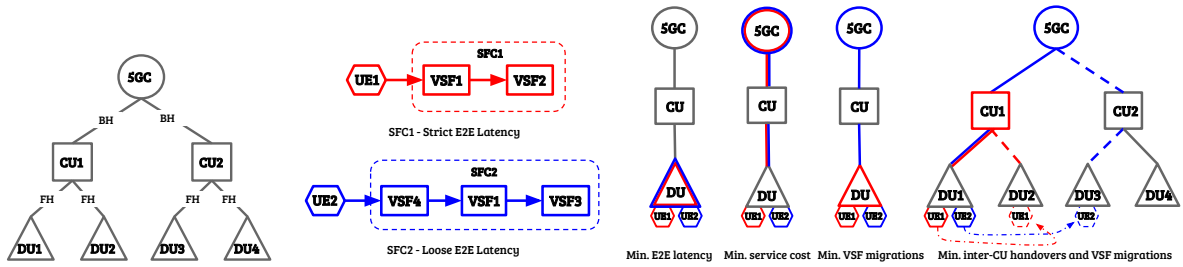
Figure 1b illustrates examples of SFCs with the red and blue requests having, respectively, strict and loose latency requirements. Receiving SFC requests by the UEs, the network provider shall associate UEs to DUs, place their requested SFCs onto the network and allocate sufficient resources (e.g., computational 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 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, SFC placement, and resource allocation can be formally stated as follows:

Given: a 5G network with the NG–RAN architecture, the computational capacity of each DC/node, 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, (iii) minimize the number of VSF migrations and their effect onto the UE's QoE, and (iv) apart from the previous objective, minimize also the number of inter-CU handovers in the network.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.3028216, IEEE Transactions on Mobile Computing



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

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

TABLE 1: Mobile network parameters

Parameter	Description
G_{net}	Mobile network graph.
N_{net}	Set of nodes/DCs in G_{net}.
N_{5gc}	Set of 5GCs in G_{net}.
N_{du}, N_{cu}, N_{ndu}	Set of DUs, CUs, non-DUs in G_{net}, respectively.
N^d_{5gc}	Core node that is connected to the DU $d \in N_{du}$.
N^d_{cu}	CU that is connected to the DU $d \in N_{du}$.
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 SFC classes.
E_{net}	Set of FH and BH links in G_{net}.
$\omega^{s,i}_{num}(n)$	# of UEs that can share instance i of VSF s on n.
$\omega^{s,i}_{prc}(n)$	Processing capacity of instance i of VSF s on n.
$\omega_{cpu}(n)$	Processing capacity of the node $n \in N_{net}$.
$\omega_{bwt}(e^{nm})$	Capacity of the link $e^{nm} \in E_{net}$.
$\delta(d)$	Coverage radius of the DU $d \in N_{du}$ (in meters).
μ_b	Big positive number.
$\Lambda_{prb}, \Lambda_{bwt}$	Per PRB and Mbps link bandwidth usage costs.
Λ^n_{cpu}	Per CPU usage cost at node $n \in N_{net}$.
Λ^{cu}_{rwd}	Reward for UEs remaining under the same CU.
$\Lambda^{u,n,s,i}_{rwd}$	Reward for UEs not changing the serving VSF.

TABLE 2: SFC request parameters

Parameter	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 $u \in N_{ue}$.
N^u_{cls}	Class of the SFC requested by the UE $u \in N_{ue}$.
$E_{req}, E_{req}(u)$	Set of all virtual links, and of the UE $u \in N_{ue}$.
$\omega^u_{bwt}(e')$	Data rate demand of link $e' \in E_{req}(u)$ of UE u.
$\omega^u_{prb}(d)$	PRB demand of UE $u \in N_{ue}$ from DU $d \in N_{du}$.
ω^u_{data}	Data (in Mbit) generated by UE $u \in N_{ue}$ per sec.

Note that the mobile network/infrastructure provider is assumed to be the same entity providing the services implemented by the SFCs. The proposed optimization approach, however, can be easily adapted to consider also the case in which these entities are different.

3.2 Mobile Network Model

Let $G_{net} = (N_{net}, E_{net})$ be an *undirected* graph modelling the mobile network, where $N_{net} = N_{du} \cup N_{cu} \cup N_{5gc}$ is the union of the set of DUs, CUs, and the 5GC. 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 $\omega_{cpu}(n)$ computational capacity expressed in terms of the number of CPUs, and a single CPU is required per VSF to be instantiated. Each instance $i \in N^s_{ist}$ of VSF $s \in N_{vsf}$ instantiated at the node $n \in N_{net}$, has capacity $\omega^{s,i}_{num}(n)$ 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 a sufficient amount of PRBs in order to meet the data rate demand for the requested services. 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 of the FH/BH link connecting the nodes n and m. Table 1 summarizes the network parameters.

3.3 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 ω^u_{data} 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:

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

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^{sfc}_{exc} is the SFC execution time computed as the summation of the execution times T^{vsf}_{exc} of its component VSFs, 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% [42], 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 another UE in DL [43]. Figure 1b illustrates examples of SFC requests while Table 2 summarizes the SFC request parameters of UEs.

It is worth to mention that both the VSF and SFC placements are performed simultaneously. This is because

TABLE 3: Binary (ξ) and continuous (ζ) variables.

Variable	Description
ξ_d^u	Indicates if UE $u \in N_{ue}$ is associated with DU $d \in N_{du}$.
$\xi_{n,i}^{u,s}$	Indicates if VSF $s \in N_{sfc}^u$ requested by UE u is served by instance $i \in N_{ins}^s$ of the same VSF type on node $n \in N_{net}$.
$\zeta_{n,i}^{u,s}$	Represents the execution time of VSF $s \in N_{sfc}^u$ requested by UE u mapped on instance i of the same VSF type on node n.
$\zeta_{n,i}^s$	Execution time of i^{th} instance of VSF $s \in N_{vsf}$ of node n.
$\xi_{n,i}^s$	Indicates if any UE uses i^{th} instance of VSF s of node n.
$\xi_e^{u,e'}$	Indicates if the virtual link $e' \in E_{req}(u)$ of the UE $u \in N_{ue}$ is mapped to the substrate link $e \in E_{net}$.
$\zeta_e^{u,e'}$	Represents the transmission time of the data on virtual link $e' \in E_{req}(u)$ of UE $u \in N_{ue}$ over substrate link $e \in E_{dc}$.
ζ_e	Represents data transmission time over substrate link e.

the VSF placement takes into account the VSF demand of all the UEs, while the SFC placement is tied with the VSF placement in order to guarantee that the end-to-end latency requirements of the SFCs are satisfied.

4 Problem Formulation

4.1 MILP Formulation

Before formulating the MILP model, we need the set of DUs that provide coverage to each UE. Considering the location $loc(u)$ of 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 computed as follows:

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

Additionally, for each UE $u \in N_{ue}$, we need to know the DCs $\tilde{\Omega}(u)$ that can host VSFs of the SFC requested by each UE. In our model, either the UE's candidate DU or the CU connected to the candidate DU, or the 5GC DC connected to the CU hosting the candidate DU can serve the UE's SFC.

Table 3 shows all binary and continuous variables used in this MILP formulation. The first objective function of the MILP formulation minimizes the E2E latency to serve SFCs.

$$\textbf{\textit{MILP-Lat}}: \quad min \Bigg(\sum_{u \in N_{ue}} \sum_{n \in N_{net}} \sum_{s \in N_{sfc}^u} \sum_{i \in N_{ins}^s} \zeta_{n,i}^{u,s} + \sum_{u \in N_{ue}} \sum_{d \in N_{du}} \Big(T_{tr}^{air}(d) + T_{prp}^{air}(u,d) + T_{prc}^{du}(d) \Big) \xi_d^u + \sum_{u \in N_{ue}} T_{prc}^{ue}(u) + \sum_{u \in N_{ue}} \sum_{e \in E_{net}} \sum_{e' \in E_{req}(u)} \Big(\zeta_e^{u,e'} + T_{prp}^{fh,bh}(e) \xi_e^{u,e'} \Big) \Bigg) \quad (3)$$

It is worth to mention that since the FH/BH links and VSFs are shared, the transmission time over the links ($\zeta_e^{u,e'} = T_{tr}^{fh,bh}$) and the VSF execution time ($\zeta_{n,i}^{u,s} = T_{exc}^{vsf}$) are calculated based on aggregated traffic demand over those links and data processing demand on the VSFs, respectively. As illustrated in Fig. 1c, this objective function results in both SFCs being placed on the DU as long as the DU has sufficient computational capacity.

The second objective function (formula (4)) 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}^n (per CPU) with the latter being much more expensive than the former ones. While Λ_{prb} and Λ_{bwt} are the same for, respectively, all DUs and links, Λ_{cpu}^n varies depending on the node $n \in N_{net}$ hosting the VSF. Specifically, the closer is the host DC to DUs, the more expensive is the CPU usage cost on that DC. This cost selection approach is justified by the fact that the edge DCs possess less computational capacity compared to the 5GC DCs. This objective function places both SFCs on the 5GC, as shown in Fig. 1c, as long as the E2E latency requirements of the SFCs are satisfied.

$$\textbf{\textit{MILP-Cost}}: \quad min \Big(\sum_{u \in N_{ue}} \sum_{d \in N_{du}} \Lambda_{prb} \omega_{prb}^u(d) \xi_d^u + + \sum_{u \in N_{ue}} \sum_{n \in N_{net}} \sum_{s \in N_{vsf}} \sum_{i \in N_{ins}^s} \Lambda_{cpu}^n \xi_{n,i}^{u,s} + + \sum_{u \in N_{ue}} \sum_{e \in E_{net}} \sum_{e' \in E_{req}(u)} \Lambda_{bwt} \omega_{bwt}^u(e') \xi_e^{u,e'} \Big) \quad (4)$$

The third objective function (formula (5)) has the goal of minimizing the number of VSF migrations and their effect on the overall UEs' QoE. This is motivated by the fact that the fewer is the number of VSF migrations for the UEs across SFC mappings, the higher is the QoE of the UEs using those VSFs since VSF migrations might result in the service interruption, degrading UEs' QoE. The VSF migrations are minimized by selecting the most appropriate DC/node for VSFs to be spawned/instantiated. Note that as opposed to *MILP-Lat* and *MILP-Cost*, the CPU usage cost $\Lambda_{cpu}^{n,cl(u)}$ in *MILP-Mig* depends not only on the DC n hosting the required VSF, but also on the service class of the SFC requested by the UE u. For example, if the UE requests an SFC that has a strict E2E latency requirement, it is cheaper to serve the SFC from a DU compared to CUs or the 5GC. Conversely, if the SFC a has loose E2E latency requirement, it is cheaper to serve the SFC at the 5GC compared to CUs and DUs. This approach effectively leads to the 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.

The minimization of the effect of the VSF migration on the UEs' QoE instead is achieved by introducing $\Lambda_{rwd}^{u,n,s,i}$, which represents a reward for the UE u for using the same i^{th} instance of the VSF s in the DC n across SFC mappings. $\Lambda_{rwd}^{u,n,s,i} = 0$ if the VSF instance serving the UE is changed; otherwise, $\Lambda_{rwd}^{u,n,s,i} > 0$. This essentially means that, in order to minimize the objective function, *MILP-Mig* tends to keep serving the UEs from the same VSF instance during their mobility when mapping new SFC requests along with the old ones. As a consequence, due to UEs mobility and new UEs making SFC requests, when there is no other way but to migrate VSFs in order to embed/re-embed the SFCs and satisfy their demands, the least utilized VSFs will be migrated in order to keep awarding as many UEs as possible with the ultimate goal of minimizing the number of UEs whose QoE will be degraded due to the VSF migrations. A placement example of this objective function is displayed in Fig. 1c, where the red and the blue SFCs, thanks their E2E latency demand, are placed, respectively, on DU and 5GC.

$$\textbf{\textit{MILP-Mig}}: \quad min \sum_{u \in N_{ue}} \sum_{n \in N_{net}} \sum_{s \in N_{vsf}} \sum_{i \in N_{ins}^s} (\Lambda_{cpu}^{n,cl(u)} - - \Lambda_{rwd}^{u,n,s,i}) \xi_{n,i}^{u,s} \quad (5)$$

Finally, the last objective function (formula (6)), apart from minimizing the number of VSF migrations and its

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.3028216, IEEE Transactions on Mobile Computing

effect on the UEs' QoE, strives also to minimize the number of inter-CU UE handovers. We remind the reader that as opposed to the intra-CU handover in which the handover is performed at the CU, the inter-CU handover requires the CU change and, therefore, involves the core network for performing a path switch procedure, which requires more network resources due to required additional signalling and may result in VSF migrations if the VSFs are instantiated on the DUs or CUs. The handover minimization is achieved by the use of Λ^{cu}_{rwd} ($\Lambda^{cu}_{rwd} > \Lambda^{u,n,s,i}_{rwd}$), which represents the reward for the UEs who stay under the same serving CU during mobility and new arrival of UEs, which triggers re-mapping of all SFCs. Thus, due to its higher reward, more priority is given by *MILP-HO* to minimizing the number of inter-CU handovers that does not necessarily minimize also the number of VSF migrations. Fig. 1c illustrates an example of SFC placements performed by this objective function. UE1 and UE2, which are initially associated with DU1, due to their mobility, move to an area that is covered by both DU2 and DU3. At this point, *MILP-HO* associates the UE1 to DU2 in order to keep it under the control of CU1 without triggering an inter-CU handover. This is because UE1 has a strict E2E latency requirement and its SFC1 is placed on CU1. UE2 instead is associated with DU3 since it has a loose E2E latency requirement and its SFC2 is placed on 5GC. Since both *MILP-Mig* and *MILP-HO* seek to prevent the UEs' QoE degradation, they are more suitable to be used for the applications, such as AR/VR, real-time monitoring and control, that have stringent QoS requirements in terms of latency, jitter, packet loss, etc.

$$\textbf{\textit{MILP-HO}}: \quad min\Big(\sum_{u\in N_{ue}} \sum_{n\in N_{net}} \sum_{s\in N_{vsf}} \sum_{i\in N^s_{ins}} (\Lambda^{n,cl(u)}_{cpu} - \\ - \Lambda^{u,n,s,i}_{rwd} - \Lambda^{cu(n)}_{rwd})\xi^{u,s}_{n,i}\Big) \quad (6)$$

All the aforementioned objective functions follow a dynamic SFC embedding strategy. In essence, this means that with the arrival of a new SFC request, all the previously embedded requests along with the new one are re-embedded. This is justified by the fact that due to the UEs' mobility some of the previously mapped UEs change their location across embeddings.

We will now detail the constraints used in this MILP formulation. Regardless of the objective function, all the constraints have to be satisfied for all UEs in order for a solution to be valid. This means that upon each embedding, all the objective functions try to admit all the UEs as long as the following constraints are satisfied. Constraint (7) ensures that each UE is associated with only one DU that belongs to its candidate set (Constraint (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 (Constraint (9)) by either the UE's host DU, or the CU connected to the host DU or by the 5GC node connected to the CU of the host DU (Constraint (10)).

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

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

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

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

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

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 (Constraint (12)).

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

A VSF instance is considered to be used as long as at least one UE is served by that instance (Constraint (13)).

$$\sum_{u\in N_{ue}} \xi^{u,s}_{n,i} - \mu_b \xi^s_{n,i} \leq 0 \ \ \forall n \in N_{net}, \ s \in N_{vsf}, \ i \in N^s_{ins} \quad (13)$$

While constraint (14) makes sure that the computational capacity of the DCs is not exceeded, where if $\sum_{u\in N_{ue}} \xi^{u,s}_{n,i} \geq 1$ then $\xi^s_{n,i} = 1$, Constraint (15) sets an upper bound on the number of UEs that can share the same VSF.

$$\sum_{s\in N_{vsf}} \sum_{i\in N^s_{ins}} \xi^s_{n,i} \leq \omega_{cpu}(n) \quad \forall n \in N_{net} \quad (14)$$

$$\sum_{u\in N_{ue}} \xi^{u,s}_{n,i} \leq \omega^{s,i}_{num}(n) \ \forall n \in N_{net}, s \in N_{vsf}, i \in N^s_{ins} \quad (15)$$

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

$$\sum_{u\in N_{ue}} \sum_{e'\in E_{req}(u)} \frac{\omega^u_{data}(e')}{\omega_{bwt}(e)} \xi^{u,e'}_e - \zeta_e = 0 \quad \forall e \in E_{net} \quad (16)$$

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

$$\mu_b \xi^{u,e'}_e + \zeta_e - \zeta^{u,e'}_e \leq \mu_b \quad \forall u \in N_{ue}, \ e \in E_{net}, \ e' \in E_{req}(u) \quad (17)$$

Thus, regardless of how much is the bandwidth requirement of virtual link $e' \in E_{req}(u)$, if this link has been mapped onto the substrate link $e \in E_{net}$ ($\xi^{u,e'}_e = 1$) then its transmission time is $\zeta^{u,e'}_e = \zeta_e$. Note that the possibility of having $\zeta^{u,e'}_e > 0$ and $\xi^{u,e'}_e = 0$ is ruled out since $\zeta^{u,e'}_e$ variable is used in all objective functions, which seeks to minimize certain costs. The arguments that contain $\zeta^{u,e'}_e$ and $\zeta^{u,s}_{n,i}$ with a small coefficient are not shown in the objective functions for the sake of simplicity.

Similarly, the execution time $\zeta^s_{n,i}$ of the i^{th} instance of the VSF $s \in N_{vsf}$ on the node $n \in N_{net}$, is computed

by Constraint (18) considering the aggregated data to be processed by that VSF.

$$\sum_{u \in N_{ue}} \frac{\omega^u_{data}(s)}{\omega^{s,i}_{prc}(n)} \xi^{u,s}_{n,i} - \zeta^s_{n,i} = 0 \quad \forall n \in N_{net},\ s \in N_{vsf},\ i \in N^s_{ins} \tag{18}$$

The following Constraint (19) ensures that if the UE $u \in N_{ue}$ uses the instance i of the VSF type $s \in N_{vsf}$ of the node $n \in N_{net}$ ($\xi^{u,s}_{n,i} = 1$) then this VSF execution time $\zeta^{u,s}_{n,i} = \zeta^s_{n,i}$ is taken into account.

$$\mu_b \xi^{u,s}_{n,i} + \zeta^s_{n,i} - \zeta^{u,s}_{n,i} \leq \mu_b \tag{19}$$
$$\forall u \in N_{ue}, \quad \forall n \in N_{net}, \quad \forall s \in N_{vsf}, \quad \forall i \in N^s_{ins}$$

Finally, Constraint (20) 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.

$$\sum_{d \in N_{du}} \Big(T^{air}_{tr}(d) + T^{air}_{prp}(d) + T^{du}_{exc}(d) \Big) \xi^u_d + T^{ue}_{prc}(u) + \\ + \sum_{e \in E_{net}} \sum_{e' \in E_{req}(u)} \Big(\zeta^{u,e'}_e + T^{fh,bh}_{prp}(e) \xi^{u,e'}_e \Big) + \\ + \sum_{n \in \bar{\Omega}(u)} \sum_{s \in N^u_{sfc}} \sum_{i \in N^s_{ins}} \zeta^{u,s}_{n,i} \leq T_{lim}(u)\ \forall u \in N_{ue} \tag{20}$$

Note that the E2E latency depends on the requested service class and, along with the other constraints, has to be satisfied by all the objective functions in order for an embedding solution to be valid. It is important to mention that the time required to find an embedding solution for the SFCs by all the algorithms does not count in the estimation the E2E service latency. This could be realized, for example, by employing machine learning (ML) techniques to predict the traffic demand [44] and the location of the users then feed them to the algorithms before starting to serve the users based on the new SFC embedding results.

4.2 Heuristic

The MILP 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 MILP algorithm takes a day on Intel Core i7 laptop (3.0 GHz CPU, 16 Gb RAM) using the ILOG CPLEX 12.8 solver to associate and serve 200 UEs making latency-sensitive SFC requests each composed of three VSFs in the network composed of 4 DUs, 2 CUs, and a 5GC. 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 (*HEU-HO*) follows the same objective of *MILP-HO* based on the results of the MILP-based algorithms reported in Section 5, in which *MILP-HO* achieves the best performance across most of the metrics. The heuristic is composed of four steps. In the first step, the heuristic initiates sfc_cls matrix for each DU to keep the count of each VSF demand per service class on that 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 DCs $cand\_vsf$ for VSFs in the UE requested SFC and populates sfc_cls matrix.

Algorithm 1: *Heuristic (HEU-HO)*

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**
    for $s \in N_{vnf}$ **do**
      $sfc\_cls(d, cl, s) \leftarrow \emptyset$
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**
        $sfc\_cls(d, N^u_{cls}, s) \leftarrow sfc\_cls(d, N^u_{cls}, s) + 1$;
  $cand\_vsf(u) \leftarrow N^d_{5gc}$;
Step 2: Find VSF placement and CPU resource allocation per node;
for $n \in N_{net}$ **do**
  $map\_cand\_vsf(n) \leftarrow \emptyset$;
for $d \in N_{du}$ **do**
  • *Strict class delay VSFs mapping* $d -> N^d_{cu} -> N^d_{5gc}$;
  • *Medium class delay VSFs mapping* $N^d_{cu} -> N^d_{5gc} -> d$;
  • *Loose class delay VSFs mapping* $N^d_{5gc} -> N^d_{cu} -> d$;
  • *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**
          if *host CU is the same* **then**
            $c_{new} \leftarrow c_{link}(p, q) + c_{node}(p) - c_{ho}(p)$;
          else
            $c_{new} \leftarrow c_{link}(p, q) + c_{node}(p)$;
          $c_{curr} \leftarrow min(c_{curr}, c_{new})$;
      $m_c(p) \leftarrow m_c(p) + c_{curr}$;
  $p \leftarrow argmin(m_c(p))$;
  $mapped(u) \leftarrow p$;
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* ;

In the second step, the algorithm considers all VSFs' demand on each DU, and the VSF instantiation starts from the VSFs that belong to the SFCs with the strict latency class towards the ones with the loose latency class. Specifically, for each VSF from the strict service latency class, the algorithm first checks if a VSF is already available on the DU. If it is not available or is available but does not have enough capacity to support the UEs' demand, it instantiates a new VSF on that DU. This process is repeated on the CU connected to the DU and then on the 5GC that is connected

to the CU, which in turn is connected to the host DU until the VSF is instantiated on one of these DC. Once it has been instantiated, the sfc_cls matrix is updated subtracting those UEs' VSF demand that are under the coverage of the DU that hosted the VSF or is connected to the DC hosting the VSF. A similar process is performed for the medium latency class and the loose latency class VSFs with the order of, respectively, CU, 5GC, DU and 5GC, CU, DU, resulting in fewer VSF migrations due to latency-aware VSF placement. At the end of this process, sfc_cls becomes a matrix 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 the DCs.

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 candidate DCs that already have the VSF instance. The UE association and its SFC placement cost encompass both the link c_{link} and the node c_{node} resource usage costs. Additionally, it takes into account the discount c_{ho} if the UE is to be associated with a DU that is under the support of the same CU of the current DU. At the end of this step, the heuristic picks the DU for the UE association that would result in the cheapest cost for the UE association and its SFC placement.

Finally, in the last step, the heuristic maps the SFC requested by the UE and allocates the required resources. Specifically, for each VSF of the UE's SFC, the heuristic computes the E2E latency on each VSF instance of each candidate DC that has the requested VSF. This is followed by checking if the VSF placement on the candidate DC 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 its 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 UEs. Considering that the number of VSF instances and SFC classes are small, the overall time complexity of this heuristic algorithm is $O(n_{ue}n_{vsf}n_{dc}n_{du}n_{link}\log n_{dc})$, where $n_{dc}, n_{links}, n_{du}, n_{vsf}$ and n_{ue} are, respectively, the number of substrate DCs, links, DUs, a UE requested VSFs and the number of UEs.

5 Evaluation

The goal of this section is to compare the presented MILP-based and heuristic algorithms. We first describe the simulation setup used in our study. We then discuss the outcomes of the numerical simulations carried out in Matlab.

5.1 Simulation Environment

The mobile network considered in the simulations is composed of 7 nodes/DCs, similar to the one depicted in Figure 1a. The 5GC is connected to the CUs through 20Gbps fiber BH links, while the CUs are connected to the DUs employing 10Gbps wireless FH links. The 5GC, 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 on a DC, it can be shared among a 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, the SFC requests arrive sequentially (i.e., one batch per minute, which corresponds to a timeslot that can be changed based on several parameters such as the number of requests, their resource demand, etc.) in batches each of which is composed of 4 UEs making SFC requests. With each batch, the algorithms try to associate all UEs (also from the previous timeslots) and serve their SFC requests. Due to the scalability issue of the MILP–based algorithms, the number of SFC batch requests is restricted to 20 (80 UEs) in each simulation run. With the arrival of a new batch of UEs making SFC requests, the UEs from the previous batches change their locations by moving in random direction with the speed selected from the set of $\{5, 25, 50\}$km/h, mimicking pedestrians, cyclist, and cars, while still keeping their SFC requirements. Each SFC consists of VSFs, whose quantity is randomly picked from the set of $\{2, 3, 4\}$, while their types are then randomly picked within 10 VSF types. VSFs in an SFC are sequentially connected, similar to the one depicted in Figure 1b. Depending on the service class, the network provider has to guarantee a certain maximum acceptable E2E latency and data rate requirements. Specifically, we consider three service classes having, respectively, $[15, 50, 100]$ms E2E latency, $[400, 200, 150]$Mbps data rate requirements, generating $[1, 5, 9]$Mbit data per second to be processed by the SFC.

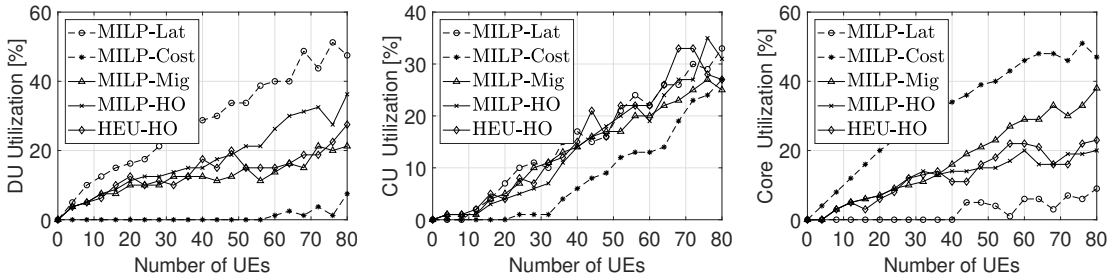
Note that like in [45], we assume that sufficient PRBs are always allocated to the UEs in order to keep high QoS and make sure that the data rate requirement of their requested SFC is always satisfied. Moreover, since the focus of this work is mostly on the SFC placement problem, the selection of a particular 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. We consider $T_{tr}^{air} = 1ms$ as a fixed transmission time interval (TTI), which corresponds to a single subframe in LTE, though TTI value can be flexibly changed in 5G [46]. $T_{tr}^{fh,bh}$ and T_{exc}^{sfc} 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 clock rate of the CPU. T_{prc}^{ue} is computed in a similar fashion for each UE. A single CPU's clock rate for 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 [47].

5.2 Simulation Results

The reported results are the average of 10 simulations with 95% confidence intervals.

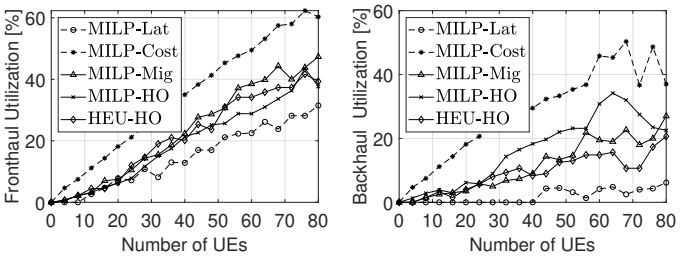
CPU utilization. Since VSFs can be shared among several UEs, and a single VSF requires one CPU to be instantiated, the CPU capacity of a DC is expressed in terms of the number of UEs that can employ VSFs/CPUs on that DC

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.3028216, IEEE Transactions on Mobile Computing



(a) CPU utilization of DUs. (b) CPU utilization of CUs. (c) CPU utilization of 5GC.

Fig. 2: CPU utilization of DU, CU and 5GC nodes.



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

Fig. 3: FH and BH link utilization.

and is equal to the number of CPUs available at the DC times the number of UEs that can share a single VSF/CPU. Consequently, the CPU utilization of a DC is computed dividing the number of UEs using VSFs of that DC by the CPU capacity of that DC.

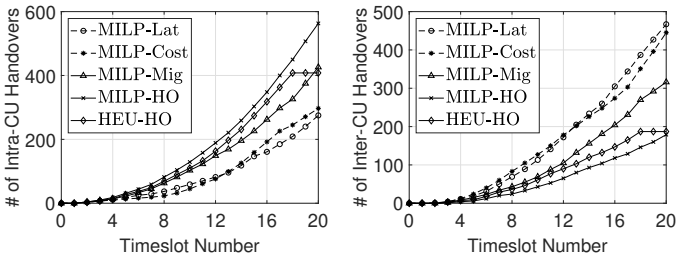
Let us now analyze the CPU utilization of DU, CU, and 5GC DCs for presented algorithms (Fig. 2) in a single simulation run. In Fig. 2a, we can observe that the CPU utilization at the DUs is the highest for the *MILP-Lat* algorithm due to the fact that, regardless of the E2E latency requirement of the requested service, *MILP-Lat* algorithm tends to instantiate VSFs at the DUs as long as they have enough CPU capacity. Conversely, the CPU utilization at the DUs is the lowest for the *MILP-Cost*. Indeed, we can observe that *MILP-Cost* starts placing VSFs at the DUs when the number of UEs in the network is 60. This can be justified by the fact that up to 56 UEs, *MILP-Cost* provisions the requested VSFs from the CUs and the 5GC. When the VSF demand increases, however, some of the VSFs are placed at DUs in order to meet the UEs' E2E latency requirements. As for *MILP-Mig*, *MILP-HO*, and *HEU-HO* algorithms, they perform a latency-aware SFC placement, as opposed to *MILP-Lat*, *MILP-Cost*, ultimately achieving similar CPU utilization that lies between the ones achieved by *MILP-Lat* and *MILP-Cost*. Thus, they do not initially tend to consume the computational resources of only DUs or the 5GC, like *MILP-Lat* and *MILP-Lat*, respectively, neglecting the latency class of the requested SFC. Note that the CPU utilization of *HEU-HO* at some DCs is lower than that of *MILP-HO* due to its lower acceptance ratio, as shown in Fig. 6b.

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 narrowed. This stems from the fact that apart from *MILP-Mig MILP-HO*, and *HEU-HO*, also *MILP-Lat* and *MILP-Cost* start serving VSFs from CUs. Specifically, *MILP-Lat* starts placing VSFs at CUs because of the lack of CPU resources at the DUs, while *MILP-Cost*, at some point, starts placing VSFs at CUs in order to satisfy the E2E latency demand of the UEs. As for CPU utilization at the 5GC (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 5GC by *MILP-Lat* algorithm up to 40 UEs. Whereas, when the number of UEs demanding SFCs increases, VSFs start being served by the 5GC due to the saturation of CPU resources at the DUs and CUs. With more UEs, the CPU utilization at the 5GC demonstrates an increasing trend for *MILP-Lat* algorithm, while its fluctuating behavior is due to UEs' mobility. In contrast to *MILP-Lat*, *MILP-Cost*, due to its cheap SFC provisioning cost, starts using the CPU resources of the 5GC right from the first batch request. Like in Fig. 2a, we can observe that the CPU utilization by *MILP-Mig*, *MILP-HO* and *HEU-HO* algorithms resembles being in between the ones achieved by *MILP-Lat* and *MILP-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 *MILP-Cost* algorithm achieves the highest level of FH and BH utilization. This is due to the fact that *MILP-Cost* strives to place VSFs at the 5GC as long as the E2E latency requirements of the SFC requested by UEs are not violated. While the FH utilization for *MILP-Cost* keeps its steady increase, the BH utilization for the same algorithm starts oscillating when more than 60 UEs are making SFC requests. As we will see in Fig. 5b, this is because of the VSF migrations from the 5GC towards the CUs, which take place in order to satisfy the E2E latency demand of all the UEs. As for *MILP-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. For what concerns the rest of the algorithms, like the CPU utilization at the DU, the FH and the BH utilization for all UEs in the cases of *MILP-Mig*, *MILP-HO* and *HEU-HO* algorithms lies between the ones achieved by *MILP-Lat* and *MILP-Cost* algorithms. Thus, compared to *MILP-Lat* and *MILP-Cost* algorithms, the latency-aware MILP–based and heuristic migration algorithms find a better compromise between the CPU utilization and the FH/BH utilization.

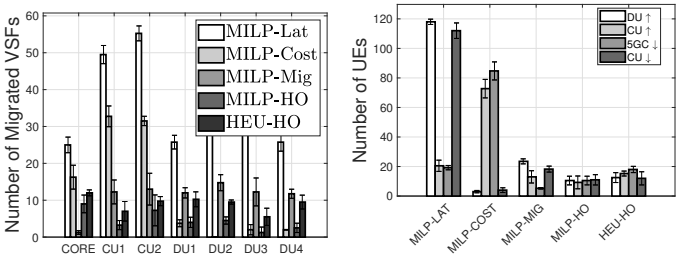
Number of handovers. One of the key differences of this study compared to our previous study [10] is that here we consider mobile UEs and propose an MILP-based and heuristic algorithms (*MILP-HO* and *HEU-HO*) that apart from minimizing the number of VSF migrations and their impact on the QoE of UEs, also minimize the number of inter-CU handovers. Let us analyze Fig. 4, which displays the aggregated number of both the intra-CU and the inter-CU handovers. As expected, the best performance in terms of the highest number intra-CU handovers and the lowest

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2020.3028216, IEEE Transactions on Mobile Computing



(a) Number of intra-CU HOs. (b) Number of inter-CU HOs.

Fig. 4: Number of intra-CU and inter-CU handovers.



(a) VSF migrations per DC. (b) UEs with changed host VSF.

Fig. 5: VSF migrations and UEs with changed host VSFs.

number inter-CU handovers is achieved by *MILP-HO*, followed by its heuristic counterpart, which triggers a similar quantity of handovers. The rest of the algorithms achieve a lower performance due to the fact that they do not aim at minimizing the number of handovers. Among these algorithms, however, *MILP-Mig* exhibits the best performance since its goal is to minimize the number of VSF migrations, which is partially achieved by trying to keep associating UEs to the DUs that are under the control of the same CU.

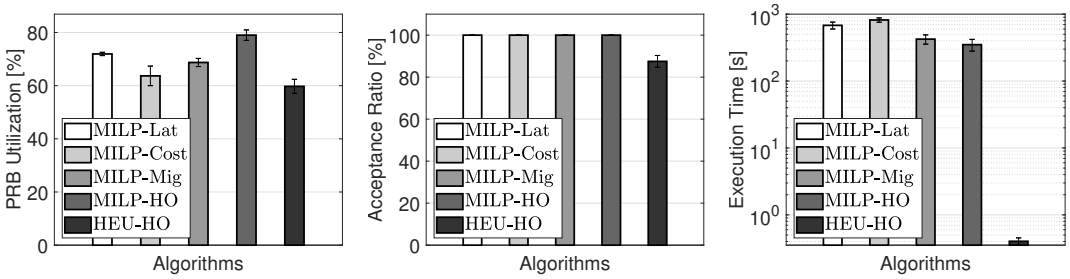
Number of VSF migrations per DC. Figure 5a illustrates the average number of aggregated VSFs migrated from each node/DC for 10 simulation runs. As expected, the highest number of VSF migrations takes place when *MILP-Lat* algorithm is used. This stems from the fact that *MILP-Lat*, regardless of the E2E latency demand of the SFCs requested by the UEs, starts instantiating VSFs as close to the UEs as possible (i.e. starting from DUs towards the 5GC), and since the CPU capacity of the DUs is much more limited compared to the CUs and the 5GC, this results in a number of migrations of those VSFs that have loose latency requirements in order to accommodate the new SFC requests with stricter latency requirements. The second-highest number of VSFs are migrated by *MILP-Cost* algorithm. This is because the migration of VSFs is triggered due to the E2E latency requirement of the requested SFC since *MILP-Cost* starts placing VSFs starting from the 5GC towards DUs, entailing high transmission delay over FH/BH links, which might result in a rejection of UEs SFC requests unless VSFs are migrated from the 5GC towards the CUs or DUs.

As for *MILP-Mig*, *MILP-HO* and *HEU-HO* algorithms, they all aim at minimizing the number of VSF migrations and their effect onto the UEs QoE, while the last two algorithms, apart from this objective also aim at minimizing the number of inter-CU handovers. We can observe that these algorithms result in more uniform VSF migrations at the DCs in contrast to *MILP-Lat* and *MILP-Cost*. We can also observe that among these algorithms, *MILP-HO* achieves the lowest amount of total VSF migrations, followed by its *HEU-HO* heuristic counterpart. This is due to the fact that these algorithms give more priority to minimizing the number of inter-CU handovers, which yields different embedding solutions from the one found by *MILP-Mig*, which, in turn, results in a lower number of VSF migrations than the one achieved by *MILP-Mig*.

VSF provisioning DC change. In order to get an insight into how the migration of VSFs takes place between different DCs, let us analyze Fig. 5b, which shows the aggregated number of UEs whose VSF provisioning DC has been changed. As expected, the highest number of host VSF changes took place when using *MILP-Lat* algorithm since this objective triggered the highest number of VSF migrations. Most of the UEs, in this case, changed their host DCs from DUs to CUs and vice versa. While the former host DC change happens due to limited CPU capacity of DUs, the latter happens mostly because of the UEs mobility whose VSFs tend to be served by the DUs in order to achieve the goal of minimizing the E2E SFC processing latency. Like in Fig. 5a, *MILP-Cost* is the second also in terms triggering the highest number of UEs host DC change with the majority of this change happening from the 5GC to CU and vice versa. While the former is the expected behavior for *MILP-Cost*, the latter occurs due to the fact that *MILP-Cost*, regardless of the SFC latency class, tends to instantiate all VSFs at 5GC ultimately saturating its CPU resources. As a consequence, some SFCs with a loose E2E latency requirement end up being instantiated at the CUs. With the arrival of more SFC requests, this results in UEs changing their host VSFs from CUs to 5GC. This can also be observed in Fig. 5a, which shows a significant number of VSF migrations by *MILP-Cost* from the CUs. As for the rest of the algorithms, compared to *MILP-Lat* and *MILP-Cost*, we can observe that they trigger much less and more uniform host DC changes across different DCs (e.g., DU-CU, CU-5GC, 5GC-CU, CU-DU). As expected, among these algorithms, the lowest number of host VSFs are triggered by *MILP-HO*, while its heuristic counterpart achieves comparable results.

PRB utilization. So far, the handover minimization algorithms (i.e., *MILP-HO*, *HEU-HO*) have demonstrated the most optimal performance in terms of CPU utilization at the DCs, FH/BH utilization, the number of both inter-CU and intra-CU handovers, and the number of VSF migrations. This performance, however, comes at the expense of the highest PRB utilization by *MILP-HO*, as can be seen in Fig. 6a, which shows the average PRB utilization for all the algorithms. This is because, in order to avoid triggering inter-CU handover for the UEs, *MILP-HO* and *HEU-HO* prefer to consume more PRBs. We can also see that the lowest PRB utilization is achieved by *HEU-HO*, which, due to its suboptimal embedding solutions, has accepted around 10% fewer requests, as shown in Fig. 6b. Among the MILP-based algorithms, the lowest PRB utilization is achieved by *MILP-Cost* since it takes into account also the cost of using PRB resources in the objective function, as opposed to MILP-Lat, for example, which prefers consuming more PRBs while keeping low the E2E latency experienced by the users.

Acceptance ratio. Since all constraints defined in Section 4.1 are imposed on all the MILP–based algorithms, although with different SFC placements due to different objective functions, they eventually accepted all the UEs with their SFC requests as shown in Fig. 6b. As opposed to the MILP-based algorithms that have always been able to find an optimal placement solution, *HEU-HO* accepted around 90% of the requests due to suboptimal SFC placements.



(a) PRB utilization. (b) Acceptance ratio. (c) Execution time.

Fig. 6: PRB utilization, 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 MILP–based algorithms. Fig. 6b shows the average execution time of associating a single SFC request for all the algorithms. It can be observed that the execution time of *HEU-HO* heuristic is three orders of magnitude less than that of the MILP-based algorithms. Thus, the heuristic algorithm proves to be much more scalable compared to the MILP-based algorithm. This scalability of the heuristic, however, comes at the expense of a lower acceptance ratio that is a consequence of suboptimal mapping solutions. Note both the heuristic and the MILP-based algorithms are centralized and, therefore, cannot be directly applied to large-scale mobile networks. Nonetheless, such networks can be divided into small clusters each of which can employ these algorithms in a centralized location (e.g., a CU).

6 Conclusions

In this study, we compared four strategies for solving a joint UE association, SFC placement, and resource allocation problem. Based on that reported results, we can conclude that *MILP-Lat*, although saves the transport network resources, is not efficient in utilizing computational resources of DUs, which are much less compared to the ones at CUs and 5GC. Seeking to minimize the E2E latency experienced by all the UEs, *MILP-Lat* tries to serve all SFC requests from the DUs, triggering the highest number of VSF migrations. Conversely, although *MILP-Cost* better utilized the computational resources of the DUs, resulting in a reduced service provisioning cost for the network provider, it has significantly increased the FH/BH link utilization. Moreover, performing a service-unaware SFC placement, due to increased transport network transmission time, it yielded many VFS migrations, especially for the UEs that have strict E2E latency requirements. While *MILP-Lat* could be an appropriate choice to be used in the network segment where the transport network lacks capacity and the edge DCs have a high processing capacity, *MILP-Cost* would be a more optimal choice in the areas where there are abundant transport network resources and the edge DCs have a limited amount of computational capacity.

As for *MILP-Mig*, it demonstrated to achieve a better performance across all evaluation metrics compared to *MILP-Lat* and *MILP-Cost* algorithms, which are two extremes in terms of employing the edge DC resources and the transport network resources. Thus, performing a service-aware SFC placement, *MILP-Mig* found a better trade-off between the computational capacity of the DCs and the FH/BH bandwidth, resulting in a much less number of VSF migrations. Among all the MILP-based algorithms, *MILP-HO* exhibited the highest performance, followed by its heuristic counterpart. Specifically, consuming slightly more PRB resources, it triggered the lowest number of inter-CU handovers and VSF migrations by optimally selecting the host DCs for the SFC requests with diverse requirements. Finally, at the expense of suboptimal UE associations and SFC placements, *HEU-HO* demonstrated the fastest execution time, making it suitable for larger-scale problems.

Acknowledgments

This work has been performed in the framework of the European Union's Horizon 2020 projects 5G-CARMEN and 5GMED co-funded by the EU under grant agreements No. 825012 and No. 951947 respectively. The views expressed are those of the authors and do not necessarily represent the project. The Commission is not liable for any use that may be made of any of the information contained therein.

References

[1] G. P. A. WG, "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] "5G; NG-RAN; Architecture description ," 3GPP TS 38.401 version 15.3.0 Release 15, Tech. Rep., 2018.
[5] 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.
[6] N. N. Katsalis, Kostas and F. R. B. T. I. Schiller, Eryk, "5g architectural design patterns," in *Proc. of IEEE ICC*, Malaysia, 2016.
[7] T. Zhao, S. Zhou, X. Guo, Y. Zhao, and Z. Niu, "Pricing policy and computational resource provisioning for delay-aware mobile edge computing." in *Proc. of IEEE ICCC*, China, 2016.
[8] B. Gao, Z. Zhou, F. Liu, and F. Xu, "Winning at the starting line: Joint network selection and service placement for mobile edge computing," in *Proc. of IEEE INFOCOM*, Paris, France, 2019.
[9] K. Qu, W. Zhuang, Q. Ye, X. S. Shen, X. Li, and J. Rao, "Delay-aware flow migration for embedded services in 5g core networks," in *Proc. of IEEE ICC*, Shanghai, China, 2019.
[10] 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.
[11] L. Tong, Y. Li, and W. Gao, "A hierarchical edge cloud architecture for mobile computing," in *Proc. of IEEE INFOCOM*, USA, 2016.
[12] 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.
[13] 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.

[14] D. Bhamare, R. Jain, M. Samaka, and A. Erbad, "A survey on service function chaining," *Journal of Network and Computer Applications*, vol. 75, pp. 138–155, 2016.

[15] Y. Xie, Z. Liu, S. Wang, and Y. Wang, "Service function chaining resource allocation: A survey," *arXiv:1608.00095*, 2016.

[16] R. Cohen, L. Lewin-Eytan, J. S. Naor, and D. Raz, "Near optimal placement of virtual network functions," in *Proc. of IEEE INFOCOM*, Hong Kong, China, 2015.

[17] J. B. Sallam, Gamal, "Joint placement and allocation of virtual network functions with budget and capacity constraints," in *Proc. of IEEE INFOCOM 2019*, Paris, France, 2019.

[18] V. Farhadi, F. Mehmeti, T. He, T. La Porta, H. Khamfroush, S. Wang, and K. S. Chan, "Service placement and request scheduling for data-intensive applications in edge clouds," in *Proc. of IEEE INFOCOM*, Paris, France, 2019.

[19] G. Sallam and J. B. Zheng, Zizhan, "Placement and allocation of virtual network functions: Multi-dimensional case," in *Proc of IEEE ICNP*, Chicago, USA, 2019.

[20] B. Wu, J. Zeng, L. Ge, S. Shao, Y. Tang, and X. Su, "Resource allocation optimization in the nfv-enabled mec network based on game theory," in *Proc. of IEEE ICC*, 2019.

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

[22] Q. Zhang, F. Liu, and C. Zeng, "Adaptive interference-aware vnf placement for service-customized 5g network slices," in *Proc. of IEEE INFOCOM*, Paris, France, 2019.

[23] S. Yang, F. Li, R. Yahyapour, and X. Fu, "Delay-sensitive and availability-aware virtual network function scheduling for nfv," *IEEE Transactions on Services Computing*, 2019.

[24] L. Wang, Z. Lu, X. Wen, R. Knopp, and R. Gupta, "Joint optimization of service function chaining and resource allocation in network function virtualization," *IEEE Access*, 2016.

[25] M. F. C. C.-F. D. S. Agarwal, Satyam, "Joint VNF Placement and CPU Allocation in 5G," in *Proc. of IEEE INFOCOM*, 2018.

[26] Q. Zhang, Y. Xiao, F. Liu, J. C. Lui, J. Guo, and T. Wang, "Joint optimization of chain placement and request scheduling for network function virtualization," in *Proc. of IEEE ICDCS*, USA, 2017.

[27] Y. Bi, C. Colman-Meixner, R. Wang, F. Meng, R. Nejabati, and D. Simeonidou, "Resource allocation for ultra-low latency virtual network services in hierarchical 5g network," in *Proc. of IEEE ICC*, Shanghai, China, 2019.

[28] D. Zhang, X. Lin, and X. Chen, "Multiple instances mapping of service function chain with parallel virtual network functions," *Journal of Algorithms & Computational Technology*, vol. 13, 2019.

[29] 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.

[30] 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.

[31] W. Li, Y. Zi, L. Feng, F. Zhou, P. Yu, and X. Qiu, "Latency-optimal virtual network functions resource allocation for 5g backhaul transport network slicing," *Applied Sciences*, vol. 9, p. 701, 2019.

[32] Q. Ye, W. Zhuang, X. Li, and J. Rao, "End-to-end delay modeling for embedded vnf chains in 5g core networks," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 692–704, 2018.

[33] R. Gouareb, V. Friderikos, and A. H. Aghvami, "Delay sensitive virtual network function placement and routing," in *Proc. of IEEE ICT*, Saint Malo, France, 2018.

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

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

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

[37] D. B. Oljira, K.-J. Grinnemo, J. Taheri, and A. Brunstrom, "A model for qos-aware vnf placement and provisioning," in *Proc. of IEEE NFV-SDN*, Berlin, Germany, 2017.

[38] D. Cho, J. Taheri, A. Y. Zomaya, and L. Wang, "Virtual network function placement: Towards minimizing network latency and lead time," in *Proc. of IEEE CloudCom*, 2017.

[39] L. W. M. Y.-G. S. Huang, Meitian, "Throughput maximization of delay-sensitive request admissions via virtualized network function placements and migrations," in *Proc. of IEEE ICC*, USA, 2018.

[40] 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.

[41] X. Zhou, B. Yi, X. Wang, and M. Huang, "Approach for minimising network effect of vnf migration," *IET Communications*, 2018.

[42] " 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.

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

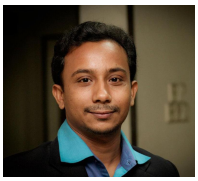
[44] T. Subramanya, D. Harutyunyan, and R. Riggio, "Machine learning-driven service function chain placement and scaling in mec-enabled 5g networks," *Computer Networks*, vol. 166, 2020.

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

[46] "5G; NG; Overall description ," 3GPP TS 38.300 version 15.4.0 Release 15, Tech. Rep., 2019.

[47] 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.

Davit Harutyunyan received his M.Sc. and Ph.D degrees (cum loude) in telecommunication engineering and information & communication technology, respectively, from the National Polytechnic University of Armenia in 2015 and from the University of Trento in 2019. He is currently an Expert Researcher in Smart Networks and Services Unit at FBK. He was the recipient of the Best Student Paper Awards of IEEE CNSM 2017 and IEEE NetSoft 2019. His main research interests include network slicing, multi-access edge computing and virtualization technologies.



Nashid Shahriar received his Ph.D degree from the School of Computer Science, University of Waterloo in 2020. He received his MSc. and BSc. degrees in Computer Science and Engineering from Bangladesh University of Engineering and Technology (BUET) in 2011 and 2009, respectively. He was a recipient of Ontario Graduate Scholarship, President's Graduate Scholarship, and David R. Cheriton Graduate Scholarship with the University of Waterloo. He received several recognitions, including the 2020 PhD Alumni Gold Medal from the University of Waterloo, the IEEE/ACM/IFIP CNSM 2019 Best Paper Award, IEEE NetSoft 2019 Best Student Paper Award, and the IEEE/ACM/IFIP CNSM 2017 Best Paper Award. His research interests include network virtualization, 5G network slicing, and network reliability.

Raouf Boutaba received the M.Sc. and Ph.D. degrees in computer science from the University Pierre & Marie Curie, Paris, in 1990 and 1994, respectively. He is currently the director of the School of Computer Science at the University of Waterloo. He is the founding editor in chief of the IEEE Transactions on Network and Service Management (2007–2010) and on the editorial boards of other journals. He received several best paper awards and recognitions including the Premier's Research Excellence Award, the IEEE ComSoc Hal Sobol, Fred W. Ellersick, Joe LociCero, Dan Stokesbury, Salah Aidarous Awards, and the IEEE Canada McNaughton Gold Medal. He is a fellow of the Royal Society of Canada, the Institute of Electrical and Electronics Engineers (IEEE), the Engineering Institute of Canada, and the Canadian Academy of Engineering. His research interests include resource and service management in networks and distributed systems.

Roberto Riggio is Senior Researcher at i2cat. Before that he was Head of the Smart Networks and Services Unit at FBK. His research interests revolve around optimization and algorithmic problems in networked and distributed systems. He has published more than 130 papers in internationally refereed journals and conferences. He has received several awards including the IEEE INFOCOM Best Demo Award (2013 and 2019) and the IEEE CNSM Best Paper Award (2015). He is a member of the ACM and a Senior Member of the IEEE.