Joint User Association and VNF Placement for Latency Sensitive Applications in 5G Networks

Rasoul Behravesh, Estefanía Coronado, Davit Harutyunyan, and Roberto Riggio
Wireless and Networked Systems, Fondazione Bruno Kessler, Trento, Italy
Email: {rbehravesh, e.coronado, d.harutyunyan, rriggio}@fbk.eu

***Abstract*—With the advent of 5G systems, telecommunication service providers (TSPs) have been facing a tremendous transition by the raised expectations of supporting billions of IoT devices and an unprecedented amount of generated data. This revolutionary transformation necessitates innovative approaches such as multi-access edge computing (MEC) to meet the requirements of many novel applications in terms of their high data rate and low latency. The idea behind MEC is to move data, virtualization, and processing capabilities from central data centers to the edge of the network. However, resources at the network edge are very scarce and costly to provision. Therefore, TSPs have to make smart decisions on how to utilize the network resources such as to make sure that the user service requirements (e.g., data rate, latency) are satisfied while the network resources are used most efficiently. In this paper, we study the problem of joint user association, VNF placement, and resource allocation, employing mixed-integer linear programming (MILP) technique. The objectives of the formulations are to minimize (i) the service provisioning cost, (ii) the number of VNF instances, and (iii) the transport network utilization, having an overarching goal of drawing a comparison between these different approaches.**

***Index Terms*—5G, MEC, NFV, placement, user association, resource allocation.**

I. Introduction

The 5th generation (5G) of cellular networks promises to transform the mobile communication landscape by providing an extremely high quality of service (QoS) for the end users. In comparison with the previous generation mobile network technologies, 5G commits to deliver sub-millisecond latency, higher connection density, multi-Gbps data rates and so forth [1]. This opens the door for many applications and services, such as augmented/virtual reality, autonomous driving, high-definition sensor sharing and so forth, which have stringent QoS requirements [2]. Nonetheless, it also calls for novel technological solutions in order to meet the requirements of such applications. Multi-access edge computing (MEC) [3] is one of such technologies that is expected to play a pivotal role in 5G networks by bringing the applications, services and processing capabilities closer to the end-users and, therefore, offloading the transport network and reducing the round-trip delay in the network. For instance, owing to the network function virtualization (NFV) technology, MEC enables mobile network components such as access and mobility management function (AMF), user plane function (UPF), application function (AF) and network functions such as firewalls, intrusion detection systems (IDS), and load balancers to be deployed at the network edge as virtualized network functions (VNFs) [4]. Moreover, MEC facilitates telecommunications service providers (TSPs) to deploy VNFs at their own premises or even develop a new revenue stream by offering their computing and networking infrastructure to vertical businesses and unleashing the power of dynamic orchestration of VNFs at their own control [3].

978-1-7281-4832-8/19/$31.00

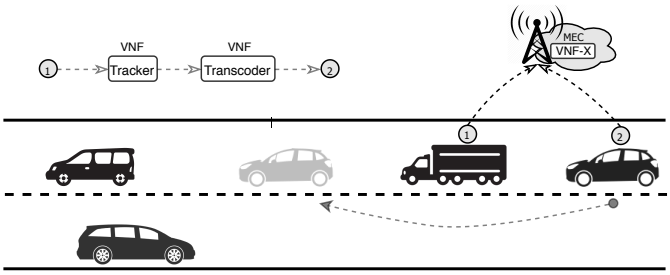


Fig. 1: An application example with a low latency requirement.

Figure 1 illustrates an example of an application, called see-through, that can take advantage of MEC nodes, which are collocated with the base stations. The figure depicts a car (number 2) being stuck behind a slow-moving truck (number 1) incapable of seeing front to check whether it is safe to pass. Therefore, the truck transmits the live video captured by forward-facing cameras to an application (composed of two VNFs, tracker and transcoder) hosted in a MEC server collocated with the base station in proximity, to be processed instantly in place and feeding to the car behind. Consequently, the car behind can see the environment blocked by the truck and based on that decide whether to pass the truck or not.

In the context of MEC, not only the edge hosts such as ordinary base stations can be endowed with computational capabilities, but also the aggregation points of the base stations (e.g., anchor base stations) and the core network. As for the cloud data centers, they could still be used for latency-tolerant applications as cheap computational resources. In general, the closer is the computing node to the user, the less is its computational capacity and the more costly is to spawn/instantiate VNFs on that node. Given the heterogeneity of computing nodes and the diversity of the QoS requirements (e.g., data rates, latency) of the application, the natural question that arises is which base stations to associate users to and where to deploy their required applications such as to make sure that their application requirements are satisfied while the network resources are used in the most efficient manner.

In this paper, we study the problem of joint user association, VNF placement, and resource allocation employing mixed-integer linear programming (MILP) technique. We con-

sider user equipments (UEs) randomly scattered in a geographical area, each requesting one service with specific demands in terms of bit rate and maximum delay tolerance. Firstly, we model the air interface delay by investigating the actual perceived air interface capacity between users and corresponding gNBs. Then, we consider three different VNF placement possibilities including (i) the MEC host collocated with gNB that the user is associated, (ii) cloud data centers, and (iii) reusing the VNF instance that has been already placed on a MEC host collocated with adjacent gNBs. To do so, we firstly consider the transmission power of gNBs and the distance of user to each gNB to achieve the air interface capacity and consequently calculating the transmission time over the link. Accordingly, based on the air interface delay the candidate gNBs for each user can be determined. Then, we develop a MILP model with three objectives to minimize (i) the service provisioning cost, (ii) the number of VNF instances, and (iii) the transport network utilization. Comprehensive simulations have been conducted to understand the efficiency of each objective in different scenarios.

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

II. Related Work

Regardless of the access technology, upon receiving the request from the user for joining the network, a mechanism is required to associate the user to the base station before transmitting data. An efficient user association mechanism leads to better resource utilization, load balancing, and energy consumption [5]. In this regard, a sizable body of literature has studied the problem of user association in 5G networks.

The study in [6] formulates the problem of user association in HetNets as a Nash bargaining problem. The objective is to maximize data rate utility while guaranteeing the minimal data requested by users and distributing the load equally among the base stations. The work in [7] presents a constrained optimization method for mobility-aware user association in mmWave networks. The method is capable of tracking the frequent variations in the network topology and channel condition, which triggers user mobility. The authors of [8] study the user association problem in a cache-enabled mobile network, capturing the trade-off between the radio network and the transport network utilization. The authors of [9] design a delay-aware user association strategy for 5G HetNets, which has been modeled as non-convex and mixed-integer nolinear programming (MINLP) problems. The main objective is to minimize overall power consumption in the network, while applying strict delay constraints. A joint user association and user scheduling solution is presented in [10], where the authors aim at minimizing the achievable throughput of the users. However, none of the aforementioned studies considers the problem of VNF placement and its effect on user's overall perceived quality of experience.

Along with the user association problem, VNF placement in the context of latency strict applications in 5G networks is the other intention of this study. Although VNF placement has been studied well, its study in the scenario of edge computing necessitates a deeper consideration. Authors in [11], introduce an integer linear programming (ILP) model to map VNFs on the servers with the goal of minimizing the number of utilized servers. The work, however, does not take into account the underlying network characteristics, just considering services and VM requests. The study in [12] investigates a VNF orchestration problem (VNF-OP), and proposes an ILP and a heuristic solution to determine the number of required VNFs and locations that they should be placed without violating service level agreements (SLA). The main objective of the work is to minimize OPEX and resource fragmentation. Another work by [13] addresses the problem of service function chain (SFC) placement with the objective of efficiently utilizing the network resources while respecting the end-to-end (E2E) latency requirement of the users.

The study in [14] jointly solves the problems of VNF placement and CPU allocation in 5G network. A queuing-based system model is proposed to consider all the entities that are involved in 5G networks. The study proposed in [15] designs an orchestration platform for jointly optimizing the VNF placement problem in three phases VNF chain composition, VNF forwarding graph embedding, and VNF scheduling. Another work by [16] studies the problem of VNF migration and instantiation. The objective is to maximize the network throughout, while satisfying the resource requirements of the services along with their E2E latency.

The closest studies to ours are [17] and [18]. The work in [17] formulates the problem of VNF placement at the network edge in order to minimize the network latency from the users to their respective VNF hosted on edge servers. A method is presented to dynamically re-schedule VNFs to attain optimal allocation and avoiding SLA violation. The study by [18] presents an ILP model to jointly solve the problems of user association, SFC placement, and resource allocation, in which users are assumed to have different E2E latency and data rate requirements. However, both of the works consider the problem of SFC placement but they lack of having a realistic model to compute the air interface delay. Moreover, as opposed to our study, they do to not consider the case in which the users may be associated with one gNB while still receive service from the VNFs that are instantiated on a MEC node collocated with neighboring gNB.

III. Network Model

A. Problem Statement

Figure 2 depicts the reference network architecture in which MEC hosts are collocated with gNBs, which are in charge of providing coverage to the users and performing their baseband signal processing. The computational capacity of the MEC hosts is very limited, which makes their usage quite costly. Conversely, the cloud data center has abundant computational resources, which makes it significantly cheaper solution to be used for instantiating VNFs compared to the MEC hosts, although it additionally requires transport network resources.

It is assumed that each user requests a service with a certain bit rate and delay tolerance. Upon receiving the service request from the user, the network provider shall make a decision on how to embed the request to the network such as to make sure that the user service requirements are satisfied, while the network resources are used in the most efficient way. Consider Fig 2 as an example of the proposed network model, in which each MEC host possess one CPU core, while an unlimited number of CPU cores is obtainable in the cloud. Fig. 2b depicts the service requests composed of users and the requested service, which are classified into latency-sensitive and latency tolerant. The VNF mapping in Fig 2c illustrates that the service requested by UE-2 is placed in the cloud, the services for UE-1 and UE-3 are mapped on the MEC hosts at the edge due to the strict latency requirement they possess. Therefore, UE-3 reuses the already deployed service on a MEC host collocated by adjacent gNB due to the unavailability of CPU resources for instantiating another VNF instance on the local MEC host.

Depending on the requirements of the services and the availability of the substrate network resources, there may be several mapping possibilities each of which corresponding to a certain objective function. The problem of joint user association, VNF placement, and resource allocation can be formally stated as follows:

Given: a 5G network composed of a set of MEC hosts collocated with gNBs, a set of links connecting MEC hosts to the transport network and then to the cloud, which also can be used for hosting services. Moreover, a set of users randomly scattered in a geographical area, each requesting a service with its respective data rate and latency requirement.

Find: joint user association, VNF placement, and resource allocation in the network.

Objective: minimize (i) the service provisioning cost, (ii) the number of VNF instances in the network, and (iii) the transport bandwidth consumption.

B. Mobile Network Model

Let $G = (N, E)$ be an undirected graph modeling the mobile network, where N represents the computing nodes, which are the union of the set of MEC hosts N_{edge} collocated with gNBs N_{gnb} (as shown in Figure 2), and the cloud node N_{cloud}. E represents the set of links connecting MEC hosts to the cloud. Each computing node $n \in N$ in the network is equipped with a certain amount of processing capacity represented by $\mathcal{C}_{cpu}(n)$. There is a link $e^{m,n} \in E$ between nodes $m, n \in N$ if they are directly connected.

Let $\omega^{i,s}_{cpu}$ represent the number of CPU cores assigned to a VNF instance, and it is assumed that at least a single CPU core is required to spawn/instantiate a VNF, while it is also possible to allocate three CPUs to a VNF instance depending on the data processing demand. The processing capacity $\mathcal{C}^{i,s}_{proc}(n)$ of instance $i \in N^s_{inst}$ of VNF $s \in N_{vnf}$ on node $n \in N$ is calculated by multiplying the number of CPU cores $\omega^{i,s}_{cpu}$ of instance $i \in N^s_{inst}$ of VNF $s \in N_{vnf}$ by the processing capacity of each CPU core. We also assume that each VNF instance upon being deployed on node $n \in N$ has a limited capacity $\mathcal{C}^{i,s}_{max}(n)$, which is expressed in terms of the maximum number of users that can be served from that VNF instance; therefore, upon reaching the limitation, further service requests for that specific VNF instance will be rejected. Additionally, each instance $i \in N^s_{inst}$ of VNF $s \in N_{vnf}$ has its corresponding throughput that is defined as $\mathcal{C}^{i,s}_{thr}(n)$ and it should not be violated. It is worthwhile to mention that we also tackle the case in which multiple instances of the same VNF are needed due to high traffic demand. Finally, each link $e^{m,n} \in E$ connecting the nodes $m, n \in N$ in the network has a certain bandwidth capacity $\mathcal{C}_{bwt}(e)$ in Gbps. Table I summarizes the parameters of the mobile network.

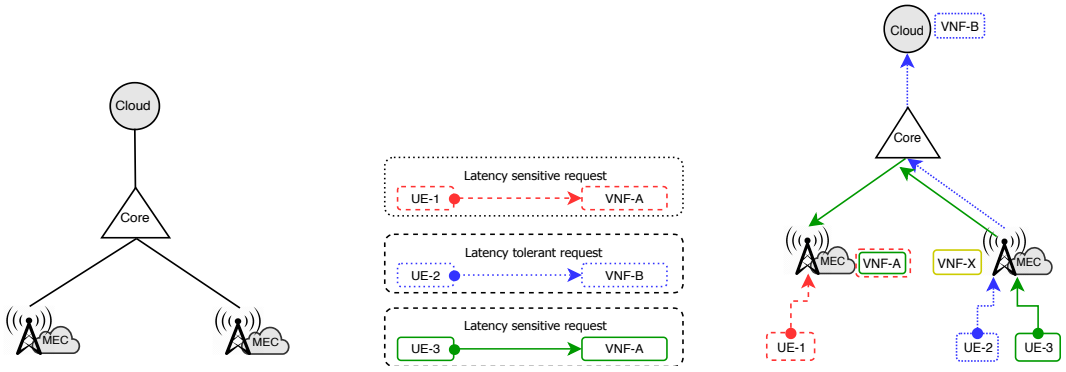
TABLE I: Mobile Network Parameters.

Parameters	Description
$G(N, E)$	Graph representing the mobile network.
N_{edge}	Set of MEC hosts in the substrate network.
N_{cloud}	Set of cloud servers.
N	Set of computing nodes in the substrate network $N = N_{edge} \cup N_{cloud}$.
E	Set of links connecting the nodes in the substrate network.
N_{gnb}	Set of gNBs in the mobile network.
N_{vnf}	Set of services.
N^s_{inst}	Set of instances of service $s \in N_{vnf}$.
$\omega^{i,s}_{cpu}$	The number of CPU cores that are required to run instance $i \in N^s_{inst}$ of service $s \in N_{vnf}$.
ξ^n_{cpu}	The cost of one CPU core on node $n \in N$.
ξ^e_{bwt}	The cost of using one Mbps bandwidth of link $e \in E$.
$\mathcal{C}^u_b$	The maximum achievable data rate between gNB $b \in N_{gnb}$ and user $u \in N_{ue}$.
$\mathcal{C}^{i,s}_{max}(n)$	The maximum number of users that can use the instance $i \in N^s_{inst}$ of service $s \in N_{vnf}$ on node $n \in N$.
$\mathcal{C}_{cpu}(n)$	The CPU capacity of node $n \in N$.
$\mathcal{C}^{i,s}_{proc}(n)$	Processing capacity of instance $i \in N^s_{inst}$ of service $s \in N_{vnf}$ on node $n \in N$.
$\mathcal{C}^{i,s}_{thr}(n)$	The maximum achievable throughput of instance $i \in N^s_{inst}$ of service $s \in N_{vnf}$ on node $n \in N$.
$\mathcal{C}_{bwt}(e)$	The bandwidth capacity of the substrate link $e \in E$.
$\mathcal{C}_{tx}(e)$	Transmission capacity of substrate link $e \in E$.
$d_{(b,u)}$	Distance between gNB $b \in N_{gnb}$ and user $u \in \bar{N}_{ue}$.
P^b_{tx}	The transmission power of gNB $b \in N_{gnb}$.
μ	A big positive number.

C. Service Request Model

We model the service requests as a directed graph $\bar{G} = (\bar{N}, \bar{E})$, where $\bar{N}$ is the union of users and their requested services, $\bar{N} = \bar{N}_{ue} \cup \bar{N}_{vnf}$, and $\bar{E}$ represents the virtual links between users and services. It is assumed that users are randomly scattered in a geographical area and each user can be associated to only one gNB.

In our model each user $u \in \bar{N}_{ue}$ requests only one service $s \in \bar{N}^u_{vnf}$, specifying the maximum delay tolerance by $T_{max}(u)$ and data rate demand ω^u_{bwt} per second that should be processed by the allocated VNF instance. The delay of a service is estimated after embedding the requested VNF instance providing the service and is computed considering the summation of the transmission time over the air, which is considered to be equal to one transmission time interval (TTI = 1ms), propagation time over the air and transport network, and the processing time of the VNF instance. It is



(a) Mobile network model. (b) Service request model. (c) VNF Mapping.

Fig. 2: Sample mobile network and service request models.

TABLE II: Service Request Model.

Parameters	Description
$\bar{G}(\bar{N}, \bar{E})$	Service request graph.
$\bar{N}$	Set of users and requested services $\bar{N} = \bar{N}_{ue} \cup \bar{N}_{vnf}$.
$\bar{N}_{ue}$	Set of users in the network.
$\bar{N}_{vnf}$	Set of services requested by the users.
$\bar{N}^u_{vnf}$	Represent service $s \in \bar{N}_{vnf}$ requested by user $u \in \bar{N}_{ue}$.
$\bar{E}$	Set of virtual links connecting users to the services.
ω^u_{bwt}	Data rate requested from user $u \in \bar{N}_{ue}$.
$T_{max}(u)$	Maximum delay tolerance of user $u \in \bar{N}_{ue}$.
$T^u_{tx}(b)$	The transmission time between user $u \in \bar{N}_{ue}$ and gNB $b \in N_{gnb}$.
$T^u_{prp}(b)$	The propagation time between user $u \in \bar{N}_{ue}$ and gNB $b \in N_{gnb}$.

worth mentioning that for the sake of simplicity, each service is represented as a single VNF instance. The problem formulation, however, can be easily adapted to support more complex service function chains. Nonetheless, it would dramatically increase the execution time of the proposed MILP-based algorithm without adding any significant value. Table II depicts the notations used for the service requests.

D. Air Interface Capacity Calculation

The air interface capacity between gNB $b \in N_{gnb}$ and user $u \in \bar{N}_{ue}$ is denoted by $\mathcal{C}^u_b$, which is a function of signal-to-interference-plus-noise-ratio (SINR) and can be computed through the following equation:

$$\forall b \in N_{gnb}, \forall u \in \bar{N}_{ue}: \quad SINR_{b,u} = \frac{P^b_{tx} d^{-\delta}_{(b,u)}}{\mathcal{N}^2 + \sum_{k \neq b} P^k_{tx} d^{-\delta}_{(k,u)}} \tag{1}$$

Where P^b_{tx} indicates the transmission power of gNB $b \in N_{gnb}$. It is worth noting that users will experience different signal strengths from the gNBs since cells are overlapping in the area of coverage. $d(b,u)$ denotes the physical distance between gNB $b \in N_{gnb}$ and UE $u \in \bar{N}_{ue}$, while δ represents the path loss coefficient and $\mathcal{N}$ is the noise power. Accordingly, if we define W as the system bandwidth, the maximum achievable air interface capacity $\mathcal{C}^u_b$ between gNB $b \in N_{gnb}$ and user $u \in \bar{N}_{ue}$ can be computed as follows:

$$\mathcal{C}^u_b = W \log(1 + SINR_{b,u}) \tag{2}$$

IV. Problem Formulation

The joint user association, VNF placement and resource allocation problem is modeled as a virtual network embedding (VNE) problem, which is NP-hard and has been studied extensively in the literature [19], [20]. The embedding process consists of two parts: the node embedding and the link embedding. In the node embedding, each virtual node (i.e., users and VNFs) in the request is mapped to a substrate node (i.e., gNBs, MEC nodes and cloud nodes in the substrate network). In the link embedding instead, each virtual link is mapped to a single substrate path. In both cases, nodes and links constraints must be satisfied.

A. MILP Formulation

Table III represents the variables used in the MILP model. As given in Formula (3), the objectives of the MILP model are to minimize (i) the service provisioning cost, (ii) the number of VNF instances in the network, and (iii) the transport network utilization. The notations ξ^n_{cpu} and ξ^e_{bwt} in Formula (3) represent the cost of using a single CPU core on the node $n \in N$ and one Mbps link bandwidth on the link $e \in E$, respectively. Λ_{vnf} and Λ_{bwt} instead represent weighting factors for VNFs and link bandwidth, respectively. Different values of these weighting factors lead to different objective functions. For example, if $\Lambda_{vnf} = \Lambda_{bwt} = 1$ then the objective would be to minimize the service provisioning cost; whereas, if $\Lambda_{vnf} = 1$, $\Lambda_{bwt} = 0$, and $\xi^n_{cpu} = 1$ this would correspond to the objective of minimizing the number of VNF instances. Finally, if $\Lambda_{vnf} = 0$, $\Lambda_{bwt} = 1$, and $\xi^e_{bwt} = 1$ the objective function would minimize the transport network utilization.

$$\begin{aligned} Minimize: & \sum_{n \in N} \sum_{s \in N_{vnf}} \sum_{i \in N^s_{inst}} \Lambda_{vnf} \xi^n_{cpu} \omega^{i,s}_{cpu} \chi^{i,s}_n \\ & + \sum_{u \in \bar{N}_{ue}} \sum_{\bar{e} \in \bar{E}} \sum_{e \in E} \Lambda_{bwt} \xi^e_{bwt} \omega^u_{bwt} \chi^{u,\bar{e}}_e \end{aligned} \tag{3}$$

We will now detail all the constraints in this MILP formulation. Regardless of the objective function, all the following

constraints have to be satisfied in order for a solution to be valid. In order to reach an optimal solution for the model, all the constraints should be satisfied. Constraint (4) enforces each user $u \in \bar{N}_{ue}$ to be associated to only one gNB $b \in N_{gnb}$, which has sufficient capacity in order to support the throughput requirement of the user (constraint (5)).

$$\forall u \in \bar{N}_{ue} : \sum_{b \in N_{gnb}} \chi_b^u = 1 \tag{4}$$

$$\forall b \in N_{gnb} : \sum_{u \in \bar{N}_{ue}} \omega_{bwt}^u \chi_b^u < \mathcal{C}_b^u \tag{5}$$

As stated before, our model assumes that each user requests only one service, which is represented as a single VNF. Thus, constraint (6) ensures that the service requested by user $u \in \bar{N}_{ue}$ is served by only one instance of the requested service type.

$$\forall u \in \bar{N}_{ue}, \forall s \in \bar{N}_{vnf}^u : \sum_{i \in N_{inst}^s} \sum_{n \in N} \chi_{u,n}^{i,s} = 1 \tag{6}$$

The following constraint guarantees that a VNF is spawned/instantiated only if at least one user is mapped on that VNF.

$$\forall n \in N, \forall s \in N_{vnf}, \forall i \in N_{inst}^s : \sum_{u \in \bar{N}_{ue}} \chi_{u,n}^{i,s} - \mu * \chi_n^{i,s} \leq 0 \tag{7}$$

Constraint (8) guarantees that a service can be instantiated on a node as long as it has sufficient amount of computational resources to host the service.

$$\forall n \in N : \sum_{s \in N_{vnf}} \sum_{i \in N_{inst}^s} \omega_{cpu}^{i,s} \chi_n^{i,s} \leq \mathcal{C}_{cpu}(n) \tag{8}$$

Similar to the node capacity constraint (8), each deployed VNF has a limitation in terms of the maximum throughput that it can support, which is enforced by constraint (9). Additionally, by constraint (10) we set an upper bound on the number of users that can use the same VNF instance.

$$\forall n \in N, \forall s \in N_{vnf}, \forall i \in N_{inst}^s : \sum_{u \in \bar{N}_{ue}} \omega_{bwt}^u \chi_{u,n}^{i,s} \leq \mathcal{C}_{thr}^{i,s}(n) \tag{9}$$

$$\forall n \in N, \forall s \in N_{vnf}, \forall i \in N_{inst}^s : \sum_{u \in \bar{N}_{ue}} \chi_{u,n}^{i,s} \leq \mathcal{C}_{max}^{i,s}(n) \tag{10}$$

Constraint (11) ensures that the virtual links can be mapped onto a substrate link as long as the link has sufficient capacity:

$$\forall e \in E : \sum_{u \in \bar{N}_{ue}} \sum_{\bar{e} \in \bar{E}(u)} \omega_{bwt}^u \chi_e^{u,\bar{e}} < \mathcal{C}_{tx}(e) \tag{11}$$

The processing time $T_{proc}^{i,s}(n)$ of the i^{th} instance of service s on the node n is computed by constraint (12) considering the aggregated data to be processed by that service instance, while constraint (13) ensures that if the user u uses that VNF instance ($\chi_{u,n}^{i,s} = 1$) then the VNF processing time $T_{proc}^{i,s}(n) = T_{proc}^{i,s}(u,n)$ is taken into account.

$$\forall n \in N, \forall s \in N_{vnf}, \forall i \in N_{inst}^s : \sum_{u \in \bar{N}_{ue}} \frac{\omega_{bwt}^u}{\mathcal{C}_{proc}^{i,s}(n)} \chi_{u,n}^{i,s} - T_{proc}^{i,s}(n) = 0 \tag{12}$$

TABLE III: Binary (χ) and continuous (T) variables.

Variables	Description
χ_b^u	Indicates if user $u \in \bar{N}_{ue}$ is associated to gNB $b \in N_{gnb}$.
$\chi_n^{i,s}$	Indicates if the instance $i \in N_{inst}^s$ of the service $s \in N_{vnf}$ is running on the node $n \in N$.
$\chi_{u,n}^{i,s}$	Indicates if user $u \in \bar{N}_{ue}$ is served by instance $i \in N_{inst}^s$ of the service $s \in N_{vnf}$ running on node $n \in N$.
$\chi_e^{u,\bar{e}}$	Indicates if the virtual link $\bar{e} \in \bar{E}$ belonging to the request by user $u \in \bar{N}_{ue}$ is mapped on the substrate link $e \in E$.
$T_{proc}^{i,s}(n)$	Processing time of instance $i \in N_{inst}^s$ of service $s \in N_{vnf}$ on node $n \in N$.
$T_{proc}^{i,s}(u,n)$	Processing time of instance $i \in N_{inst}^s$ of service $s \in N_{vnf}$ on node $n \in N$ for user $u \in \bar{N}_{ue}$.
$T_{tx}(e)$	Transmission time over link $e \in E$.
$T_{tx}^{u,\bar{e}}(e)$	Transmission time over link $e \in E$ for virtual link $\bar{e} \in \bar{E}$.

$$\forall n \in N, \forall u \in \bar{N}_{ue}, \forall s \in N_{vnf}, \forall i \in N_{inst}^s : \mu * \chi_{u,n}^{i,s} + T_{proc}^{i,s}(n) - T_{proc}^{i,s}(u,n) \leq \mu \tag{13}$$

A similar approach is adopted by constraint (14) to compute the transmission time $T_{tx}(e)$ over the substrate link e, while constraint (15) handles the accurate transmission time computation over the virtual link $\bar{e}$. Note that in both constraints (12) and (14), ω_{bwt}^u refers to the amount of data generated by the user $u \in \bar{N}_{ue}$.

$$\forall e \in E : \sum_{u \in \bar{N}_{ue}} \sum_{\bar{e} \in \bar{E}(u)} \frac{\omega_{bwt}^u}{\mathcal{C}_{tx}(e)} \chi_e^{u,\bar{e}} - T_{tx}(e) = 0 \tag{14}$$

$$\forall e \in E(u), \forall \bar{e} \in \bar{E}, \forall u \in \bar{N}_{ue} : \mu * \chi_e^{u,\bar{e}} + T_{tx}(e) - T_{tx}^{u,\bar{e}}(e) \leq \mu \tag{15}$$

Constraint (16) makes sure that for each virtual link there will be a continues path established between the gNB the user is associated with and the node hosting the requested service.

$$\forall i \in N, \forall e^{n,m} \in \bar{E}(u), \forall u \in \bar{N}_{ue} : \sum_{e \in E^{n\rightarrow}} \chi_e^{e^{n,m}} - \sum_{e \in E^{\rightarrow n}} \chi_e^{e^{n,m}} = \begin{cases} -1 & \text{if } i = n \\ 1 & \text{if } i = m \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

Where $E^{n\rightarrow}$ represents the links originating from node $n \in N$ and $E^{\rightarrow n}$ includes the links entering node $n \in N$.

The delay of a service $s \in N_{vnf}$ is computed from the time the request is issued until the time the requested data is received by the user. We consider the propagation delay, transmission delay, and the computing delay for each user $u \in \bar{N}_{ue}$, where the propagation and transmission delay is composed of the air interface delay, and the transport link in the case the VNF is spawned/instantiated on the cloud node. Constraint (17) guarantees that the aggregated delay does not exceed the maximum delay budget defined for the user u:

$$\forall u \in \bar{N}_{ue} : \sum_{n \in N} \sum_{s \in N_{vnf}} \sum_{i \in N_{inst}^s} T_{proc}^{i,s}(u,n) + \sum_{e \in E} T_{tx,prp}^{u,\bar{e}}(e) + \sum_{b \in N_{gnb}} T_{tx,prp}^{u}(b) \leq T_{max}(u) \tag{17}$$

V. Evaluation

This section provides a deep comparison between the three objectives of the proposed MILP-based model. As stated earlier, the objectives of the formulations are to minimize (i) the service provisioning cost, (ii) the number of VNF instances, and (iii) the transport network utilization. In this regard, we first present the simulation environment following by a discussion on the numerical results taken from Gurobi mathematical optimization solver [21].

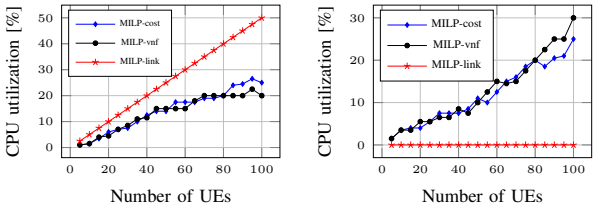
It should be noted that the weight coefficients Λ_{vnf} and Λ_{bwt} in the objective function (formula (3)) decide which objective the model is following. Accordingly, when $\Lambda_{vnf} = \Lambda_{bwt} = 1$ then the objective would be to minimize the service provisioning cost; whereas, if $\Lambda_{vnf} = 1$, $\Lambda_{bwt} = 0$, and $\xi^n_{cpu} = 1$ this would correspond to the objective of minimizing the number of VNF instances. Finally, if $\Lambda_{vnf} = 0$, $\Lambda_{bwt} = 1$, and $\xi^e_{bwt} = 1$ the objective function would minimize the transport network utilization.

A. Simulation Environment

A mobile network composed of 11 nodes out of which one is cloud and the others are edge hosts (edge nodes are collocated with gNBs) is considered. The connection from gNBs to the core is established through 100 Mbps links. Also, the link from the core to the cloud is supposed to be the aggregation of all the links originating from gNBs and ending at the core. The cloud node and edge nodes have, respectively 100 and 2 CPU cores, and each core has 3.4GHz clock rate. We also assumed that spawning a new VNF instance requires at least one CPU core available on the node. Once a VNF is instantiated on a node, it can be shared among the maximum of 10 users under the condition of not violating the E2E latency of the users connected to the VNF instance. Every minute, which is considered a single time slot, a new batch composed of 5 users each of which making a service request is arriving. Upon receiving the service requests, the algorithms try to serve the users and find the best solution depending on the objective function to associate the users to the gNBs, place the VNFs on the edge hosts or at the cloud, and allocate enough resources to the spawned VNF. We consider 20 batches of service requests, which results in 100 requests. Three different service classes are assumed with 6 VNF types having strict, medium, and loose E2E latency. The maximum latency in the range for strict, medium and loose latencies are [20, 70, 200] milliseconds, respectively. Depending on the VNF class, the network provider has to guarantee a certain E2E latency and data rate requirement for the user.

B. Simulation Results

CPU Utilization. As stated before, running one VNF instance requires at least one CPU core available on the node. Also, we indicated that VNFs can be shared among the maximum of 10 users. Therefore, the CPU capacity of a node is expressed as the number of users that can use a single VNF, times the number of CPU cores available on that node. Accordingly, the CPU utilization on a node is equal to the number of users using the VNFs on that node divided by the overall CPU capacity of the node. Figure 3a depicts



(a) CPU utilization of edge nodes. (b) CPU utilization of cloud.

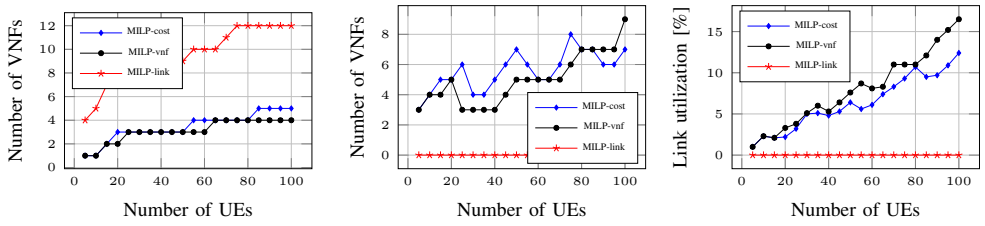
Fig. 3: CPU utilization of edge and cloud nodes.

CPU utilization of the edge nodes as a function of users for three different algorithms. As it can be inferred, MILP-link algorithm achieves the highest CPU utilization because it aims at minimizing bandwidth consumption, resulting in all the VNFs being placed on the MEC nodes collocated with gNBs. As for MILP-cost and MILP-vnf algorithms, they resemble in terms of CPU utilization, the latter exhibits slightly better performance, which is due to its objective, which leads to many users using the same VNF instance, avoiding instantiation of new VNFs instances.

Number of VNFs. In order to gain an insight into how many VNFs are placed on the cloud and edge nodes, let us analyze Fig. 4a and Fig. 4b, which show the number of VNFs at the edge and cloud nodes, respectively. As demonstrated in Fig. 4a, MILP-vnf spawns a fewer number of VNFs compared to MILP-cost and MILP-link algorithms. The rationale behind this performance comes from the fact that MILP-vnf does not consider the cost of links and its only intention is to minimize the number of VNF instances, while MILP-cost prefers to minimize both link and CPU cost. We can see from Fig. 4a that MILP-cost also greatly decreases the number of placed VNFs on the substrate network. The reason behind this result is that MILP-cost has the objective of minimizing the service provisioning cost, which means CPU cost has a huge impact on the overall cost of service provisioning. Therefore, instantiation of more VNFs results is higher CPU consumption and consequently increasing the service provisioning cost.

Link Utilization. Figure 4c illustrates the link utilization versus the number of users for all the proposed algorithms. As it can be observed, the link utilization for the MILP-link algorithm always remains zero in all the iterations due to the placement of all the VNFs on the edge nodes. The reason stems from the fact that unless resources are available at the edge it always prefers to use them. Therefore, we can comprehend that the algorithm uses the cloud resources starting from the time that there is no resource at the edge hosts. As for the other algorithms MILP-cost achieves a lower bandwidth utilization compared to MILP-vnf algorithm. This originates from the fact that as opposed to MILP-vnf algorithm, MILP-cost algorithm takes the link bandwidth usage cost into account.

Execution time. Regarding the time required to solve the embedding request composed of 20 batches (100 users) making service requests, it takes 16, 12, and 10 seconds for MILP-link, MILP-cost, and MILP-vnf algorithms, respectively. The problem becomes computationally intractable when we consider larger substrate networks and more complex



(a) VNFs at the edge nodes. (b) VNFs at the cloud. (c) Link utilization.

Fig. 4: Number of VNFs embedded on edge and cloud nodes and link utilization.

service requests composed of multiple VNFs. In future works, we intend to develop heuristic algorithms to address the scalability issue of the proposed MILP-based algorithms.

VI. Conclusion

This paper proposed a novel joint user association, VNF placement, and resource allocation strategy for latency-critical applications in the context of 5G networks employing mixed-integer linear programming technique. We proposed three objectives to minimize (i) the service provisioning cost, (ii) the number of VNF instances, and (iii) the link utilization. The obtained results demonstrated the outperformance of MILP-cost algorithm comparing to the other algorithms in terms of CPU utilization that is due to the importance of CPU cost parameter in the objective function. As opposed to the first objective, MILP-vnf algorithm, which tries to minimize the number of VNF instances, achieved better results in terms of the number of embedded VNF instances both on the MEC hosts and the cloud. While MILP-link cannot take advantage of the abundance of cheap resources available in the cloud, it minimizes the bandwidth utilization by prioritizing edge hosts over the cloud servers. Although the results confirmed the zero usage of cloud resources for the MILP-link algorithm, we claim that by increasing the number of users and termination of resources at the edge, the algorithm begins to embed VNFs on the cloud servers as well. For future work, we intend to study the performance of the algorithms in a real scenario with a much larger number of users. Furthermore, investigating the problem of VNF migration, which happens by user mobility or computing resource termination of the MEC hos in the scenario of edge computing is an interesting area of research that necessitates deeper investigations to be performed. Moreover, we aim to tackle the scalability issue of the proposed methods by proposing a heuristic algorithm, which can reach a near-optimal solution in a considerably shorter time scale.

Acknowledgments

This work has been performed in the framework of the European Union's Horizon 2020 project 5G-CARMEN co-funded by the EU under grant agreement No 825012. 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] A. Gupta and R. K. Jha, "A survey of 5G network: Architecture and emerging technologies," *IEEE Access*, vol. 3, pp. 1206–1232, 2015.
[2] G. Group *et al.*, "View on 5g architecture," *White Paper, July*, 2016.
[3] S. Kekki, W. Featherstone, Y. Fang, P. Kuure, A. Li, A. Ranjan, D. Purkayastha, F. Jiangping, D. Frydman, G. Verin *et al.*, "MEC in 5G networks," *ETSI, White Paper*, 2018.
[4] S. Abdelwahab, B. Hamdaoui, M. Guizani, and T. Znati, "Network function virtualization in 5G," *IEEE Communications Magazine*, vol. 54, no. 4, pp. 84–91, 2016.
[5] D. Liu, L. Wang, Y. Chen, M. Elkashlan, K.-K. Wong, R. Schober, and L. Hanzo, "User association in 5G networks: A survey and an outlook," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1018–1044, 2016.
[6] D. Liu, Y. Chen, K. K. Chai, and T. Zhang, "Nash bargaining solution based user association optimization in HetNets," in *Proc. of IEEE CCNC*, Las Vegas, NV, USA, 2014.
[7] A. S. Cacciapuoti, "Mobility-aware user association for 5G mmWave networks," *IEEE Access*, vol. 5, pp. 21 497–21 507, 2017.
[8] D. Harutyunyan, A. Bradai, and R. Riggio, "Trade-offs in Cache-enabled Mobile Networks," in *Proc. of IEEE CNSM*, Rome, Italy, 2018.
[9] Y. Lei, G. Zhu, C. Shen, Y. Xu, and X. Zhang, "Delay-Aware User Association and Power Control for 5G Heterogeneous Network," *Mobile Networks and Applications*, vol. 24, pp. 1–13, 2018.
[10] X. Ge, X. Li, H. Jin, J. Cheng, and V. C. Leung, "Joint user association and user scheduling for load balancing in heterogeneous networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 5, pp. 3211–3225, 2018.
[11] H. Moens and F. De Turck, "VNF-P: A model for efficient placement of virtualized network functions," in *Proc. of IEEE CNS*, San Francisco, CA, USA, 2014.
[12] M. F. Bari, S. R. Chowdhury, R. Ahmed, and R. Boutaba, "On orchestrating virtual network functions," in *Proc. of IEEE CNSM*, Barcelona, Spain, 2015.
[13] 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.
[14] S. Agarwal, F. Malandrino, C.-F. Chiasserini, and S. De, "Joint VNF placement and CPU allocation in 5G," in *Proc. of IEEE INFOCOM*, Honolulu, HI, USA, 2018.
[15] 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*, vol. 4, pp. 8084–8094, 2016.
[16] 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.
[17] R. Cziva, C. Anagnostopoulos, and D. P. Pezaros, "Dynamic, latency-optimal VNF placement at the network edge," in *Proc. of IEEE INFOCOM*, Honolulu, HI, USA, 2018.
[18] 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.
[19] M. Chowdhury, M. R. Rahman, and R. Boutaba, "Vineyard: Virtual network embedding algorithms with coordinated node and link mapping," *IEEE/ACM Transactions on Networking*, vol. 20, no. 1, pp. 206–219, 2012.
[20] A. Fischer, J. F. Botero, M. T. Beck, H. De Meer, and X. Hesselbach, "Virtual network embedding: A survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 1888–1906, 2013.
[21] "Gurobi mathematical optimization solver," Accessed on 20.06.2019. [Online]. Available: https://www.gurobi.com/