Optimum Network Slicing for Ultra-reliable Low Latency Communication (URLLC) Services in Campus Networks

Iulisloi Zacarias, Francisco Carpio, André Costa Drummond, Admela Jukan
Institut für Datentechnik und Kommunikationsnetze
Technische Universität Braunschweig
Braunschweig, Germany
{i.zacarias, f.carpio, andre.drummond, a.jukan}@tu-braunschweig.de

***Abstract*—Within 3GPP, the campus network architecture has evolved as a deployment option for industries and can be provisioned using network slicing over already installed 5G public network infrastructure. In campus networks, the ultra-reliable low latency communication (URLLC) service category is of major interest for applications with strict latency and high-reliability requirements. One way to achieve high reliability in a shared infrastructure is through resource isolation, whereby network slicing can be optimized to adequately reserve computation and transmission capacity. This paper proposes an approach for vertical slicing the radio access network (RAN) to enable the deployment of multiple and isolated campus networks to accommodate URLLC services. To this end, we model RAN function placement as a mixed integer linear programming problem with URLLC-related constraints. We demonstrate that our approach can find optimal solutions in real-world scenarios. Furthermore, unlike existing solutions, our model considers the user traffic flow from a known source node on the network's edge to an unknown *a priori* destination node. This flexibility could be explored in industrial campus networks by allowing dynamic placement of user plane functions (UPFs) to serve the URLLC.**

I. Introduction

The fifth generation of cellular networks (5G) has fostered innovation in the industrial world, making it possible to transmit data at very low latencies and high reliability. The so-called vertical slicing of radio access network (RAN) is essential to creating public-network integrated campus networks, which are especially attractive in Industry 4.0 [1]. To this end, the base stations can be split into the three functional blocks: radio unit (RU), distributed unit (DU), and centralized unit (CU). With virtualization and *softwarization* of these radio functions, we can create dedicated and isolated RAN, which are, in effect, the provisioned campus network [2]. To guarantee the reliability, the network operator must ensure proper slice resource provisioning while dealing with failures or varying service requirements [3]. To this end, the redundancy of resources and the isolation between slices is essential to guarantee steady performance.

This work investigates the efficient usage of network slicing (NS) to provision dedicated RAN for campus networks requiring high isolation for ultra reliable low latency communication (URLLC) type of industrial environments. We consider a fully disaggregated RAN composed of RU, virtual CU, and virtual DU, whereby each functional block can be placed individually in the network [2]. The proposed approach addresses the use of RAN slices in a logical isolation setup by deploying dedicated network functions (NFs) for each slice [4]. As a deployment scenario, a multi-tiered network is considered. The radio functions can be placed in small data centers at the network's edge, aiming for low latency (i.e., a URLLC slice). The core cloud can be explored for services with less stringent requirements, where computing resources are abundant and the computation delays are reduced. The main idea is to make the least strict applications to be computed closer to the core while reserving edge computation availability for URLLC. Efficient algorithmic solutions to optimize the slice allocation, as the one addressed in this paper, are still an open challenge towards feasible URLLC services.

We formulate the problem as a mixed integer linear programming (MILP) aiming to find the best trade-off between the average computation load of servers and the traffic load of logical links while guaranteeing the requirements of URLLC services. We demonstrate that our approach can find optimal solutions in real-world scenarios. Unlike existing solutions, our model considers the user traffic flow from a known source node on the network's edge to an unknown *a priori* destination node since its location will depend on where the CU is placed. This flexibility could be explored in industrial campus networks by allowing dynamic placement of user plane functions (UPFs) to meet the URLLC requirements. Our novel contribution is also in joint consideration of computation and networking delays.

The rest of this paper is organized as follows: Section II presents the related work. Section III describes the reference network and the optimization model. The scenario evaluated is presented in Section IV. Section V presents the results. Section VI concludes the paper.

II. Related Work

When implementing campus networks with network slicing, efficient placement of RAN functions plays an essential role.

978-1-6654-7598-3/23/$31.00 ©2023 IEEE

Disaggregated 5G RAN function placement using general purpose hardware is investigated in [2], [4], [5], [6], and [7]. The authors investigated the problem from different perspectives. For example, [4] aim at energy efficiency when placing RAN functions. [6] focus on minimizing the economic cost of RAN and multi-access edge computing (MEC) deployments while maximizing the served traffic. [2] finds the trade-off between maximum aggregation level (*i.e.,* maximum amount of grouped RAN NFs) and minimum computing resources.

Our approach is similar to [2], [4] and [5], which consider a multi-tier network with varying computing resources at each tier. The slicing problem of RAN is tackled in [5], [6], and [7]. Although the authors consider different types of network slices (URLLC, enhanced mobile broadband (eMBB), and massive machine-type communication (mMTC)) and their varying and sometimes contrasting requirements regarding network capacity and latency, a fixed latency is attributed to queueing, and processing of traffic by RAN functions. Since the requirements for the same type of slices can vary to some extent (*e.g.,* URLLC slices with different bandwidth requirements), these slices can experience different end-to-end (E2E) service delays affecting the final quality of service (QoS) of the slice.

To contribute to the existing body of knowledge, this work applies MILP to solve the problem of vertical slicing RAN by allowing the sharing of public infrastructure among slices with strict and, at times, competing requirements (e.g., URLLC and eMBB slices sharing the same RAN). Our approach solves this problem in real-world multi-tiered topologies, just like the previously mentioned work. However, unlike previous work, we consider the delays introduced by processing and queuing at the servers in addition to network delays, which, depending on the size of the demand, can largely impact the total delay experienced by the service. In this way, we consider the delay caused by the processing of one slice to other slices sharing the same server, guaranteeing that the traffic load applied to one slice does not cause service-level agreement (SLA) violations in other slices, which is critical to URLLC performance.

III. Campus Network Slicing

This section presents the reference industrial campus network scenario and the characterization of applications supported in typical industrial premises. After that, the formal system optimization model is presented.

A. Reference Network Scenario

We consider a scenario in which a campus network is deployed for an industry vertical hosted by the public network, i.e., public-network integrated deployment [1]. The public mobile network operator uses network slicing to deploy the campus network according to the requirements, including reliability and security.

Fig. 1 illustrates the scenario. We assume that a single campus network, composed of two Production lines connected by separate base band units (BBUs), can require different slices depending on its applications' requirements. In Production line #1, a slice of type URLLC is required to provide connectivity to applications supporting the augmented workers. This application must provide real-time, high-quality video rendering while reacting to the worker's head movements. Thus it requires low latency, high reliability, and high bandwidth. The second slice of type URLLC is required to connect the autonomous robots in Production line #2. This application also requires low latency and high reliability to ensure the correct functioning of the robot safely, however with less stringent requirements in terms of bandwidth. Moreover, both Production lines need one more slice each for the equipment temperature monitoring service, being those slices of type mMTC. Finally, this campus network comprises four 5G public network slices.

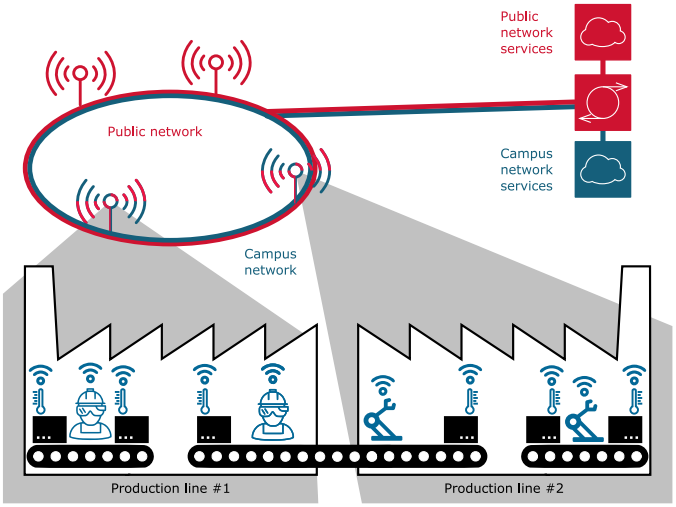

Fig. 1. Deployment scenario of a public network integrated campus (non-public) network in a factory plant composed of several production lines connected by multiple base stations. Adapted from [1].

Since the slices composing the campus network share the RAN with the public network and other slices, vertical network slicing is employed to guarantee the requirements of the URLLC slices. Furthermore, by employing the virtualization of the NFs, the mobile network operator can provide a dedicated RAN for each slice, ensuring the isolation among slices with competing requirements or among slices belonging to different industries. The slice isolation is achieved by providing a dedicated radio protocol stack employing virtualization of RAN functional blocks for each application.

The next-generation RAN (NG-RAN) proposed for 5G implements a decoupled radio protocol stack allowing the splitting of base stations into up to three functional blocks: RU, DU, and CU. Additionally, softwarization and virtualization of radio functions can reduce the cost of RAN deployments by placing each functional block individually in the virtualized network on top of general-purpose hardware [2].

The campus network services, represented in Fig. 1 as a blue cloud box, are provided by the instantiation of a set of isolated virtual RANs, each composed of a separate NF chain (RU, DU, and CU). Fig. 2 shows examples of possible allocations of NFs to provide virtual RANs in a multi-tier network with

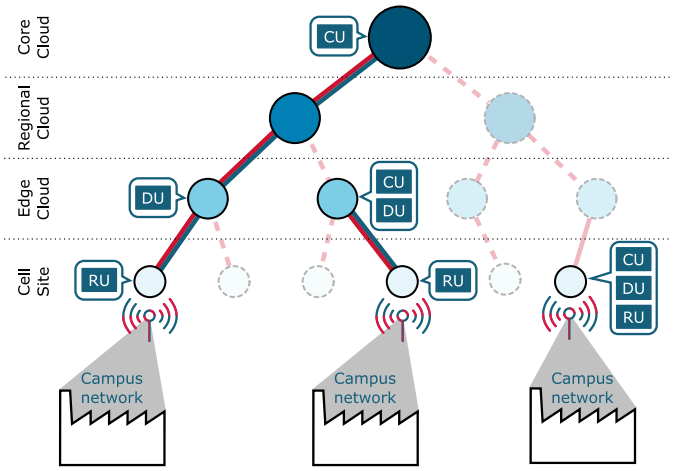

Fig. 2. Three examples of virtual RANs allocation in a multi-tier network.

computational nodes layered according to its proximity to the BBUs. The cell site (CS) is the closest location where a NF can be placed hosting RUs providing radio interface to user equipments (UEs), but also the one with the least amount of computational resources. On top of that, three levels of cloud computing nodes can be considered, with the core cloud having the larger computational capacity. On the other hand, there are far more instances of CSs compared to core cloud sites. Three different slices are presented as part of separate campus networks. The slices' configuration differs from each other considering link and computation availability, resource cost, and the service requirements; *e.g.,* an URLLC slice will need a shorter NF chain, with DU and CU allocated closely to the BBU. Therefore, the closer the NF to the BBU, the lower the network delay. At the same time, nodes closer to the edge have fewer computing resources, which leads to higher computational delays. Therefore, a trade-off between network and computational resource allocation needs to be addressed to meet the requirements of URLLC services.

B. System Model

The network is modeled as a graph $\mathbb{G} = (\mathbb{N} \cup \mathbb{X}, \mathbb{L})$ where $\mathbb{N} = \{1, \ldots, N\}$ is a set of sites hosting a set of servers $\mathbb{X} = \{1, \ldots, X\}$ and a set $\mathbb{L} = \{1, \ldots L\}$ of directed links connecting the sites. The delay of connection among servers placed on a site (i.e., internal data center connections) is disregarded in this work. Each link $l \in \mathbb{L}$ connecting two sites has limited capacity regarding bandwidth (given in Mbps) and also a propagation delay (given in fraction of a second), represented by C_l^{max} and D_l respectively. The set $\mathbb{S} = \{1, \ldots, S\}$ denotes all RAN slice instances provisioned in the network, which can be implemented as a collection of NFs composing a virtualized network service [2]. Therefore, a specific slice instance $s \in \mathbb{S}$ corresponds to an ordered set of NFs denoted by $\mathbb{V}_s = \{1, \ldots, V_s\}$ where the NF $v \in \mathbb{V}$ is the v^{th} NF in the chain composing the RAN slice. Each NF $v \in \mathbb{V}$ has its type $t \in \mathbb{T}$, which defines the parts of the radio protocol stack that the NF supports. Since NFs are not shared among slices, we guarantee the isolation of each slice, even when sharing the same server. Without loss of generalization, we assume that each campus network is composed of a subset of one or more slices $s \in \mathbb{S}$.

The traffic demands were modeled having the CS as its source, and the last NF in the set $\mathbb{V}_s$ (*i.e.,* the CU) as its destination. Hence the destination node of the traffic demand can vary, since the placement of the CU depends on the model decision. We assume that each UE is connected to a single RU placed at one of the CS. The input set $p \in \mathbb{P}_s$ is the set of k-shortest paths from the source node in the edge to its corresponding CU that can be placed in any intermediate node between the CS and the Core Cloud sites (including these sites). All parameters considered in the problem formulation are summarized in Table I, while the variables used in the MILP model are summarized in Table II.

TABLE I
LIST OF PARAMETERS USED IN THE MODEL

Parameter	Meaning
$\mathbb{N}$	set of sites: $N = \{1, \ldots, N\}, n \in N$.
$\mathbb{X}$	set of servers: $X = \{1, \ldots, X\}, x \in X$.
$\mathbb{L}$	set of links: $L = \{1, \ldots, L\}, l \in L$.
$\mathbb{P}$	set of admissible paths: $P = \{1, \ldots, P\}, p \in P$.
$\mathbb{S}$	set of slices: $S = \{1, \ldots, S\}, s \in S$.
$\mathbb{T}$	set of NF types: $T = \{\text{RU}, \text{DU}, \text{CU}\}$.
$\mathbb{V}_s$	ordered set of NFs, $V_s = \{\text{RU}_s, \text{DU}_s, \text{CU}_s\}$.
$\mathbb{\Lambda}$	set of traffic demands: $\mathbb{\Lambda} = \{\lambda, \ldots, \mathbb{\Lambda}\}$.
$\mathbb{\Lambda}_s \subset \mathbb{\Lambda}$	subset of traffic demands $\lambda \in \mathbb{\Lambda}$ for slice $s \in S$.
$\mathbb{N}_p \subset \mathbb{N}$	subset of ordered sites in path $p \in P$.
$\mathbb{X}_n \subset \mathbb{X}$	subset of servers attached to site N.
$\mathbb{X}_p \subset \mathbb{X}$	ordered subset of servers composing path p.
$\mathbb{P}_s \subset \mathbb{P}$	subset of admissible path $p \in P$, for $s \in S$.
T_p^l	binary, 1 if path $p \in P$ traverses link $l \in L$.
$T_p^{n,m}$	binary, 1 if path $p \in P$ connects node $n \in N$ and $m \in N$.
$\Gamma_{t(v)}^{pro}$	continuous, load ratio of a NF $v \in V$ of type $t \in T$.
C_l^{max}	integer, maximum capacity of the link $l \in L$.
C_x^{max}	integer, maximum capacity of server $x \in X$.
$C_{x,t(v)}^{\text{proq, max}}$	integer, maximum processing capacity that can be assigned by a server x to a NF of type t
D_l	continuous, propagation delay of link $l \in L$.
D_s^{max}	continuous, maximum service delay of a slice $s \in S$.
$D_{t(v)}^{pro,max}$	continuous, maximum allowed processing delay for a NF of type t.
$D_{t(v)}^{pro,min}$	continuous, minimum processing delay for a NF of type t.
$D_{t(v)}^{proq}$	continuous, delay of a NF v of type t due to queueing.
$D_{t(v)}^{prox}$	continuous, delay of a NF v of type t due to processing.

TABLE II
LIST OF VARIABLES USED IN THE MODEL

Variable	Meaning
z_p^s	binary, 1 if slice s uses path $p \in P_s$.
$z_p^{\lambda,s}$	binary, 1 if traffic demand λ from slice s uses path $p \in P_s$.
f_x	binary, 1 if server x is used.
$f_x^{v,s}$	binary, 1 if NF v from slice s is allocated at server x.
$f_{x,\lambda}^{v,s}$	binary, 1 if NF v from slice s is used at server x by traffic demand λ.
$d_p^{\lambda,s}$	continuous, service delay of a traffic demand λ in a path p.
k_x	continuous, utilization cost of server $x \in X$.
k_l	continuous, utilization cost of a link $l \in L$.
u_x	continuous, utilization of server $x \in X$.
u_l	continuous, utilization of a link $l \in L$.

C. Objective function

The problem formulation has the objective of minimizing the cost of all servers and links in the network, *i.e.,*

$$minimize: \frac{\alpha}{|\mathbb{X}|}\sum_{x\in\mathbb{X}} k_x + \frac{1-\alpha}{|\mathbb{L}|}\sum_{\ell\in\mathbb{L}} k_\ell$$

where k_x is the utilization cost of servers and k_ℓ is the utilization cost of links. The α parameter allows us to select the desired trade-off between the server and link cost utilization. The utilization cost of a server is defined in (1), where the variable $f_{x,\lambda}^{v,s}$ specifies if a NF v uses the server x for processing a demand $\lambda \in \mathbb{\Lambda}_s$ from a slice s. If the variable is 1, the traffic from λ, multiplied by the corresponding load ratio of NF v is added to the server utilization. The cost is normalized by the maximum capacity of server x.

The link utilization cost is defined in (2). The variable $z_p^{\lambda,s}$ defines whether a traffic demand λ from a slice s is using path p and T_p^l checks if path p contains link l in order to sum the traffic demand from l to the link utilization. Similarly to (1), the link utilization is normalized by dividing the current link utilization by the total capacity of the link. In both equations (1) and (2), a_i and b_i are terms of a piecewise linearization function.

$$k_x = a_i\left[\frac{1}{C_x^{\max}}\sum_{s\in\mathbb{S}}\sum_{v\in\mathbb{V}_s}\left(\Gamma_{t(v)}^{\text{pro}}\sum_{\lambda\in\mathbb{\Lambda}_s}\lambda\cdot f_{x,\lambda}^{v,s}\right)\right] - b_i, \forall x\in\mathbb{X}, \quad (1)$$

$$k_\ell = a_i\left[\frac{1}{C_\ell^{\max}}\sum_{s\in\mathbb{S}}\sum_{p\in\mathbb{P}_s}\sum_{\lambda\in\mathbb{\Lambda}_s}\lambda\cdot T_p^\ell\cdot z_p^{\lambda,s}\right] - b_i, \forall \ell\in\mathbb{L} \quad (2)$$

subject to a set of constraints that are described next.

D. Constraints

The following two equations enable the routing of demands from the CS to the destination CU node, which can be placed in any node of the path. For each demand $\lambda \in \mathbb{\Lambda}_s$ of slice s, exactly one path $p \in \mathbb{P}_s$ must be selected, as represented by the constraint (3). The binary variable $z_p^{\lambda,s}$ indicates that a traffic demand $\lambda \in \mathbb{\Lambda}_s$ of slice s is using the path $p \in \mathbb{P}_s$. Equation (4) takes the activated path from $z_p^{\lambda,s}$ and activates the same path for the corresponding slice instance. The left side forces z_p^s to be 1 when at least one demand of slice s is using path p while the right part forces $z_p^s = 0$ when no demands of slice s are using path p.

$$\sum_{p\in\mathbb{P}_s} z_p^{\lambda,s} = 1, \forall s\in\mathbb{S},\ \forall\lambda\in\mathbb{\Lambda}_s \quad (3)$$

$$z_p^{\lambda,s} \le z_p^s \le \sum_{\lambda'\in\mathbb{\Lambda}_s} z_p^{\lambda',s}, \forall s\in\mathbb{S}, \forall p\in\mathbb{P}_s, \forall\lambda\in\mathbb{\Lambda}_s \quad (4)$$

The next three constraints are responsible for activating NFs along the path and mapping traffic demands to DUs and CUs in an isolated fashion, for providing higher reliability. Equation (5) assures that each traffic demand from a slice is processed by every NF (*i.e.,* RU, DU, CU) in one specific server. The binary variable $f_{x,\lambda}^{v,s}$ assumes value 1 if traffic demand $\lambda \in \mathbb{\Lambda}_s$ is using a NF $v \in V_s$ from slice s at server $x \in \mathbb{X}$.

$$\sum_{x\in\mathbb{X}} f_{x,\lambda}^{v,s} = 1,\ \forall s\in\mathbb{S}, \forall v\in\mathbb{V}_s, \forall\lambda\in\mathbb{\Lambda}_s \quad (5)$$

The constraint (6) takes the activated NF for each demand from the previous equation and activates the NF for the slice that the demand belongs to. The left side of the inequality forces $f_x^{v,s}$ to assume the value 1 when at least one traffic demand $\lambda \in \mathbb{\Lambda}_s$ is using a NF $v \in \mathbb{V}_s$ at server $x \in \mathbb{X}$ whilst the right side forces $f_x^{v,s}$ to be 0 when no traffic demand $\lambda \in \mathbb{\Lambda}_s$ is using that specific NF $v \in \mathbb{V}_s$ at a specific server $x \in \mathbb{X}$.

$$f_{x,\lambda}^{v,s} \le f_x^{v,s} \le \sum_{\lambda'\in\mathbb{\Lambda}_s} f_{x,\lambda'}^{v,s}, \forall s\in\mathbb{S}, \forall v\in\mathbb{V}_s, \forall x\in\mathbb{X}, \forall\lambda\in\mathbb{\Lambda}_s \quad (6)$$

To determine wheter a server is used, the binary variable f_x is constrained according to (7). The left side of the inequality sets the value of f_x to 1 when at least one NF v of any slice $s \in \mathbb{S}$ is allocated at server $x \in \mathbb{X}$ while the right part forces f_x to be 0 if the server is not being used by any NF.

$$\frac{1}{|\mathbb{S}||\mathbb{V}_s|}\sum_{s\in\mathbb{S}}\sum_{v\in\mathbb{V}_s} f_x^{v,s} \le f_x \le \sum_{s\in\mathbb{S}}\sum_{v\in\mathbb{V}_s} f_x^{v,s},\ \forall x\in\mathbb{X} \quad (7)$$

The number of instances of NF $v \in \mathbb{V}_s$ belonging to a slice $s \in \mathbb{S}$ that can be activated in a server is limited by the constraint described in (8). This constraint limits the number of instances of a specific NF $v \in \mathbb{V}_s$ from slice $s \in \mathbb{S}$ to 1, allowing a maximum of one instance of $\mathbb{V}_s = \{\text{RU}, \text{DU}, \text{CU}\}$ being allocated for each slice.

$$\sum_{x\in\mathbb{X}} f_x^{v,s} \le 1, \forall s\in\mathbb{S}, \forall v\in\mathbb{V}_s. \quad (8)$$

Constraint (9) maps all the NFs in the set $\mathbb{V}_s$ on the activated path for slice $s \in \mathbb{S}$ forcing every NF $v \in \mathbb{V}_s$ to be instantiated in any server $x \in \mathbb{X}_p$ placed along the path $p \in \mathbb{P}_s$ for the demands $\lambda \in \mathbb{\Lambda}_s$. When $z_\lambda^{s,p}$ is 1, at least one instance of each specific NF $v \in \mathbb{V}_s = \{\text{RU}, \text{DU}, \text{CU}\}$ must be activated. On the other hand, when $z_\lambda^{s,p}$ is 0, no NFs can be allocated on servers placed along the path.

$$\sum_{x\in\mathbb{X}_p} f_{v,s}^{x,\lambda} \ge z_p^{\lambda,s}, \forall s\in\mathbb{S}, \forall p\in\mathbb{P}, \forall\lambda\in\mathbb{\Lambda}_s, \forall v\in\mathbb{V}_s \quad (9)$$

Because each slice is composed of an ordered set of NFs (*i.e.,* RU, DU, CU), the traffic demand has to traverse all these NFs in the correct order enforced by constraint (10). The left part of the inequation guarantees the ordering of NFs and it is activated only when $z_p^{\lambda,s}$ is 1, meaning that path $p \in \mathbb{P}_s$ is

activated. Therefore, for each traffic demand λ of slice s, the v^{th} NF is only allocated at server $x \in \mathbb{X}_n$ if the previous NF $v^{\text{th}-1}$ is allocated in any server $y \in \mathbb{X}_m$, where m ranges from 1 to the n traversed site in path p. The ordering of servers $x \in \mathbb{X}_n$ in site n is not modeled in our approach, and we assume that local routing policies will enforce the correct ordering inside these subsets.

$$\left(\sum_{m=1}^{n}\sum_{y\in\mathbb{X}_m} f_{y,\lambda}^{(v-1),s}\right) - \sum_{x\in\mathbb{X}_n} f_{x,\lambda}^{v,s} \geq z_p^{\lambda,s} - 1 \begin{cases} 1 < v \leq |\mathbb{V}_s| \\ n \neq m \end{cases}$$
$$\forall s \in \mathbb{S}, \forall \lambda \in \mathbb{\Lambda}_s, \forall p \in \mathbb{P}_s, \forall v \in \mathbb{V}_s, \forall m, n \in \mathbb{N} \quad (10)$$

Since RUs run part of radio frequency (RF) protocol, they should always be placed close to the RF antennas and the user equipment. The constraint (11) assures their placement at the first site composing the path, therefore always placing these type of NFs at the CS.

$$\sum_{x\in\mathbb{X}_p} f_x^{v,s} = 1, \ \forall s \in \mathbb{S}, n = 0, v = 0, p = 0. \quad (11)$$

The parameter $D_s^{\max}$ specifies the total service delay of a slice. Therefore, the E2E user plane delay from the RU to the CU is considered, being the placement of the CU dependent on the model's decision. The E2E service delay depends on the propagation delay of traffic demands when traversing links, plus the processing delay of NFs in servers, given by (12a). In (12b), the total processing load of a NF in the server x is controlled by the binary variable $f_{x,\lambda}^{v,s}$. Hence, if $f_{x,\lambda}^{v,s}$ is 1, a fraction of the queueing delay $D_{t(v)}^{\text{proq}}$ is accounted. The delay term given in (12c) adds the load independent minimum delay of a NF depending on its type $t \in \mathbb{T}$ and a varying delay part that linearly increases with the server utilization, which is calculated acording to (13).

$$d_{x,v,s}^{\text{pro}} = d_{x,v,s}^{\text{proq}} + d_{x,v,s}^{\text{prox}}, \forall s \in \mathbb{S}, \forall v \in \mathbb{V}_s, \forall x \in \mathbb{X}_p \quad (12a)$$

$$d_{x,v,s}^{\text{proq}} = D_{t,v}^{proq} \frac{\Gamma_{t(v)}^{pro} \cdot \sum_{\lambda \in \not\succeq_s} f_{x,\lambda}^{v,s}}{C_{x,t(v)}^{\text{proq, max}}} \quad (12b)$$

$$d_{x,v,s}^{\text{prox}} = D_{t(v)}^{pro,min} \cdot F_x^{v,s} + D_{t(v)}^{prox} \cdot u_x \quad (12c)$$

$$u_x = \frac{1}{C_x^{\max}} \sum_{s\in\mathbb{S}} \sum_{v\in\mathbb{V}_s} \left(\Gamma_{t(v)}^{\text{pro}} \sum_{\lambda\in\mathbb{\Lambda}_s} \lambda \cdot f_{x,\lambda}^{v,s} \right), \forall x \in \mathbb{X} \ , \quad (13)$$

Finaly, the total E2E delay of the slice on the selected path $p \in \mathbb{P}_s$ is given in (14) and it is constrained by the maximum service delay of the slice $D_s^{\max}$. The first term is the propagation delay of demands on links, which is controlled by T_p^ℓ and it is 1 only when the link is used by path p. The second term add the processing delay of the NF in a server $x \in \mathbb{X}_p$, which is controlled by the variable $f_{x,\lambda}^{v,s}$.

$$\sum_{\ell\in\mathbb{L}} D_\ell \cdot T_p^\ell + \sum_{x\in\mathbb{X}_p}\sum_{v\in\mathbb{V}_s} d_{x,v,s}^{\text{pro}}(\vec{\lambda}) \cdot f_{x,\lambda}^{v,s} \leq D_s^{\max}$$
$$\forall s \in \mathbb{S}, \forall \lambda \in \mathbb{\Lambda}_s, \forall p \in \mathbb{P}_s \quad (14)$$

E. First-fit algorithm (heuristics)

A first-fit algorithm (heuritics) is proposed for comparison purposes. Based on the O-RAN specifications [8], the algorithm realizes the function placement by ordering the set of all slice requests by the required E2E delay latency, starting with the lowest first. Sequentially, for each slice demand, the set of all admissible paths (paths containing the source and destination nodes of the demand) is selected. Next, the paths are filtered depending on the slice service type. For slices of type URLLC, only paths containing nodes with available servers at the CS or Edge Cloud are selected. For slices of type eMBB, first, we try to select paths that contain nodes with servers available at the Regional Cloud. If no servers are available, paths with servers at the Edge Cloud are selected. If still no servers are available, paths containing sites and servers at the CS are selected. For slices of type mMTC, paths containing servers at the Core Cloud are selected. In the final step, each RAN block composing the slice is placed in the first available server of the first path in the set of paths.

The proposed algorithm does not consider the maximum E2E service delay of the slice to allocate the RAN blocks. Because each new NF allocated at a server changes the processing delay of the already allocated NFs at that server, such a feature requires a complete state space search, what was already done by MILP model. The time complexity of the first-fit algorithm is $O(|\mathbb{S}|((|\mathbb{\Lambda}_s| \cdot |\mathbb{P}_s|) + |\mathbb{V}_s|))$. Note that $\mathbb{P}_s$ and $\mathbb{V}_s$ are sets with small cardinality regardless of the network scenario.

IV. Evaluation Scenario

The evaluation scenario was defined based on real network topologies currently deployed to support RAN [2], [9]. As depicted in Fig. 3, four classes of nodes compose the scenario, and the nodes are classified into tiers. CS (Tier-0) nodes host the RU of the RAN, providing radio interface for UEs of the campus network being the source of each demand. The CSs are physically located near the industrial premises where the campus network should be deployed. Multiple CSs collect mobile application data and feed this data to a smaller number of Edge Clouds (Tier-1), either directly or forwarding the information to other CS sites forming an access ring. The Edge Cloud nodes can host other blocks of the RAN (such as DUs and CUs) or be used as transport nodes. Multiple Edge Clouds are connected to fewer Regional Clouds (Tier-2), which in turn, are connected to a small number of Core Cloud sites (Tier-3).

All the sites located in a given tier have similar processing capacity, which is given in processing units, and the number of servers. The characteristics of each node regarding processing capacity can be found in Fig. 3 (top table). The considered network is composed in total by 51 nodes connected through

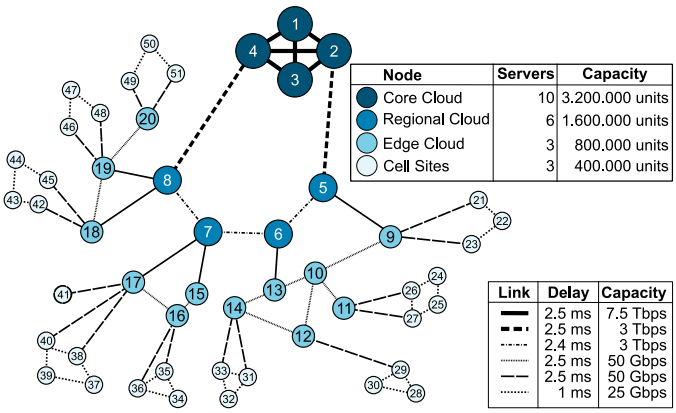

Fig. 3. Network topology used in the evaluation.

70 bidirectional links characterized by varying capacity and propagation delay according to the tiers that each link connects. The transmission delay and bandwidth of links were derived from deployments of a real metro-area 5G C-RAN [10] and the values assumed per various link types are also specified in Fig. 3 (bottom table).

Each slice deployed in the campus network is composed of a particular set of RAN NFs. The origin of slices is the RU hosted in one of the CSs of the topology depicted in Fig. 3. Revisting Fig. 1, the network slices in Production line #2 have a different source (a different RU, possibly hosted in a different CS), therefore the proposed approach considers them as two independent RAN slices. Multiple slices are created for each service type [11] with a subset of slices with the same source node composing a campus network.

Table III shows the eight slice types considered in the experiments. URLLC_RO1 and URLLC_RO2 aim to support communication with high reliability and low latency requirements, such as autonomous robots. URLLC_AW1 and URLLC_AW2 are URLLC slices but with improved data rates to support communication for augmented worker devices. Improved experience in mobile broadband services with high data rates is supported by eMBB1 and eMBB2 slice types. Finally, mMTC1 and mMTC2 slice types support machine communication with high device density and relaxed requirements regarding latency and network throughput. Therefore, in our simulated scenario, each mMTC demand is the aggregated demand of a massive number of IoT devices deployed in the industry premises. Although the number of devices is big, the device cycle time and small payload sizes characterize this service type. The values for required bandwidth and total service follow the specifications from [12], and [13].

RAN nodes have different processing requirements depending on the parts of the RAN protocol stack that its processes. The processing requirements of each NF were derived from [2]. Based on the parts of the RAN protocol that each NF process, a load ratio ($\Gamma^{pro}_{t(v)}$) is assigned according to Table IV.

The set of pre-calculated paths used to route the traffic demands from its origin in the CS to the destination node was generated from all nodes in Tier-0 (CS) towards all nodes in Tier-3 (Core Cloud). However, the model allows the last NF of each RAN slice (*i.e.,* CU) to be placed in any node along the path.

TABLE III
SLICE TYPES AND ITS REQUIREMENTS

Slice type	Bandwidth	Total service delay	RAN isolation
URLLC_RO1	4 Mbps	1 ms	✓
URLLC_RO2	25 Mbps	1 ms	✓
URLLC_AW1	100 Mbps	1 ms	✓
URLLC_AW2	1 Gbps	1 ms	✓
eMBB1	10 Gbps	4 ms	–
eMBB2	20 Gbps	4 ms	–
mMTC1	1 Mbps	15 ms	–
mMTC2	2 Mbps	15 ms	–

TABLE IV
PROCESSING CHARACTERISTICS OF EACH NF TYPE

NF type	RAN Protocols	Load ratio
CU	RRC PDCP	0.9
DU	High RLC Low RLC High MAC Low MAC High PHY	1.44
RU	Low PHY	2.16

V. RESULTS AND VERIFICATION

In evaluating the proposed approach, we compare the results of the MILP with those from the first-fit algorithm, named in the graphs as LP and HEU, respectively. Both are run considering a public network scenario with the topology depicted in Fig. 3. All the traffic demands of slices have as their source a set of nodes at the CS. As explained previously, we account for a slice's propagation and processing delay from the source of the demand until the respective CU of the RAN transporting the data. The source node for each slice is selected from a uniform distribution over all CS nodes. The type of service that each slice should support is also selected from a uniform distribution, considering the services and characteristics from Table III. The number of slices supported in the network varies from 5 to 50. The model was implemented in Java language using OpenJDK version 19 and solved with Gurobi Optimizer version 9.5.2. On average, for the highest load (50 slices), each linear programming (LP) execution took 13 minutes to complete on an Intel Core i5-10600 CPU at 3.3GHz with 16 GB of RAM with Fedora 37 Workstation. The independent replication method was considered to generate the confidence intervals, with ten executions for each plot point, for a 95% confidence level. Finally, we assume $\alpha = 0.98$ for all experiments, thus applying significantly more weight to the computation to encourage cheaper computing costs.

Fig. 4 shows the average server utilization of the RAN blocks (RU, DU and CU). As the first-fit approach could not fulfill the requirements of service delay of slices when placing more than 25 slices in the networks, these results were not considered in the evaluation. On the other hand, the LP model was able to distribute the load through the network more efficiently, placing all the slice requests in the network. The model seeks to minimize the computation cost by placing the RAN NFs in higher tiers of the network (where computational cost is reduced). However, between 30 and 40 slices, due to the topology, we can observe a slight change in the line slope caused by a shortage of computing resources in the regional cloud sites, placing the NFs of slices with a limited delay budget on the edge cloud and CSs.

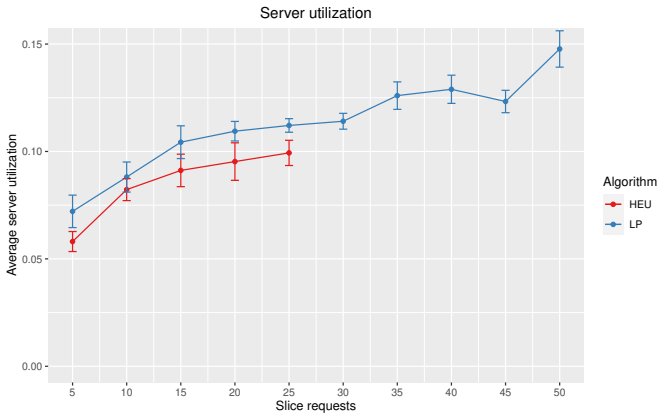

Fig. 4. Average server utilization by RAN NFs.

The average link utilization represented in Fig. 5 corroborates this information showing a reduction in link usage in the region between 30 and 40 slices. Since the NFs are being placed in lower tiers of the network (close to UEs in the cell site), the average link utilization is reduced. Due to stringent requirements regarding service delay for URLLC and eMBB service types, hindering the use of computing resources at the Core Cloud. Consequently, mobile network operators should increase the computing resource offered at the edge when planning to support many delay-sensitive industrial applications.

The delay per service type for the heuristics algorithm plotted in Fig. 6 presents slight variations due to the characteristic of the algorithm that places the RAN NFs in specific layers according to the slice type. Here it is important to recall that only placements fulfilling the delay budget of the slice are considered in this graph. In other words, placements that do not fulfill the maximum service delay (D_s^{max}) of the slice are discarded. When comparing the service delay of URLLC slices of the heuristics algorithm with the service delay of URLLC slices of the LP model on Fig. 7, a slight increase in the service delay can be observed. The LP model can load balance the processing load over servers in the network, and this behavior is expected. Surprisingly, the service delay of mMTC slices is, on average 3.93, a much smaller value than the maximum

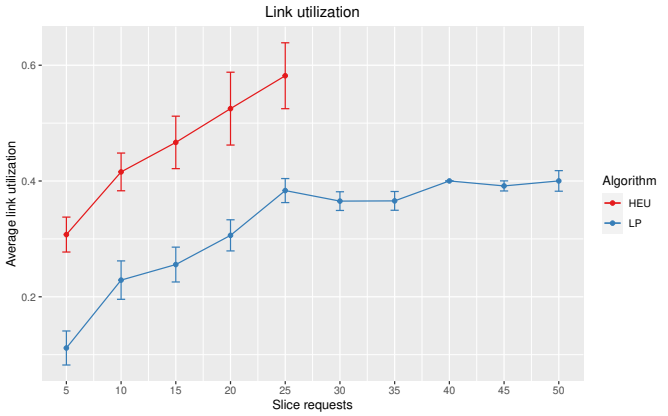

Fig. 5. Link utilization.

service delay allowed for this type of service. Since this type of slice does not require high traffic demands and the processing cost in servers is small, the LP model allocated DUs and CUs of this slice type at the regional cloud and the edge cloud without affecting URLLC and eMBB slices due to the balance between the cost of processing and cost of links.

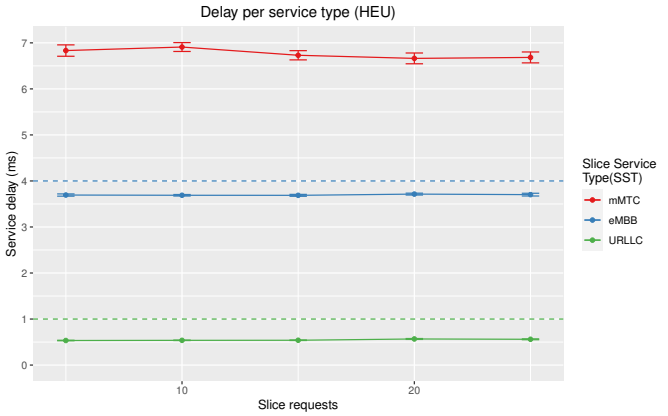

Fig. 6. Service delay experienced by the application when placing the NFs using the heuristics algorithm. Horizontal dashed lines indicate the delay budgets.

The distribution of DUs and CUs among different network tiers can be observed in Fig. 8. Due to the service delay requirements of URLLC slices, the corresponding RAN blocks are hosted in the CS and at the Edge Cloud, showing the importance of moving the computing resources to the edge when supporting industrial applications of this type. Moreover, 54.5% of the URLLC slices are completely hosted at the CS, such an allocation pattern simplifies the RAN replication in an eventual fault tolerance configuration.

Finally, as mentioned earlier in Section II, the fact that our model considers the delays introduced by processing and queuing at the server can clearly be seen in Fig. 8. Slices requiring more network capacity are placed in tiers closer to the user due to higher processing delays, thus reducing the maximum achievable distance between the UEs and CUs.

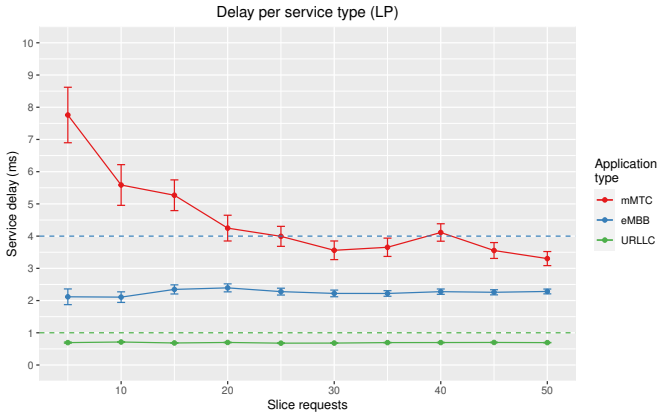

Fig. 7. Service delay experienced by the application when placing the NFs using the LP model. Horizontal dashed lines indicate the delay budgets.

The regional cloud is mainly occupied by DUs and CUs belonging to eMBB slices due to the E2E delay and high traffic demand of this type of slice. On the other hand, moving mMTC RAN blocks to the edge does not significantly impact the performance of other slices sharing the same physical resources, given its low resource requirements.

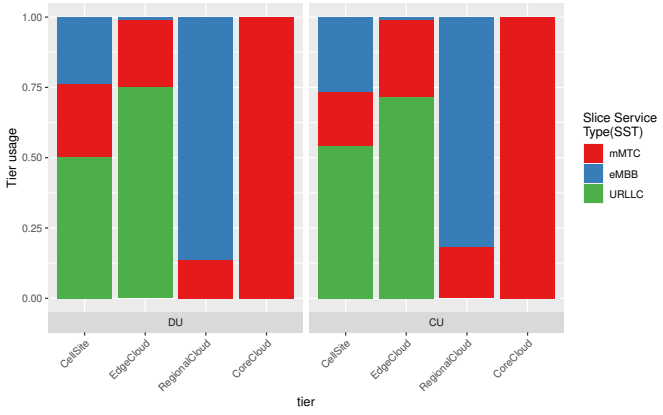

Fig. 8. Utilization of servers in each tier by DUs and CUs.

VI. Conclusions and Future Work

We proposed an approach for vertical slicing of the radio access network (RAN) to enable the deployment of multiple and isolated campus networks to accommodate URLLC services. To this end, we modellled RAN function placement as a mixed integer linear programming problem with URLLC-related constraints. Furthermore, unlike existing solutions, our model considered the user traffic flow from a known source node on the network's edge to an unknown *a priori* destination node. This flexibility could be explored in industrial campus networks by allowing dynamic placement of UPFs to serve the URLLC.

For future work, we plan to study the isolation of slices at server and node levels. Also, replication of NF could be explored to improve the reliability of NG-RAN when supporting applications requiring URLLC. We also plan to extend the model to allow the sharing of CU and DU NFs to assess the impact on reliablity.

Acknowledgment

The authors acknowledge the financial support by the Federal Ministry of Education and Research of Germany in the programme of "Souverän. Digital. Vernetzt." Joint project 6G-RIC, project identification number: 16KISK031, and the European Union's Horizon Europe under Grant Agreement no. 101096342 (HORSE project).

References

[1] "5G non-public networks for industrial scenarios," 5G Alliance for Connected Industries and Automation, Frankfurt, Germany, Tech. Rep., Jul. 2019.

[2] F. Z. Morais, G. M. F. De Almeida, L. L. Pinto, K. Cardoso, L. M. Contreras, R. d. R. Righi, and C. B. Both, "PlaceRAN: Optimal placement of virtualized network functions in beyond 5G radio access networks," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2022.

[3] R. L. Gomes, L. F. Bittencourt, and E. R. M. Madeira, "Reliability-aware network slicing in elastic demand scenarios," *IEEE Communications Magazine*, vol. 58, no. 10, pp. 29–34, 2020.

[4] H. Yu, F. Musumeci, J. Zhang, M. Tornatore, and Y. Ji, "Isolation-aware 5G RAN slice mapping over WDM metro-aggregation networks," *Journal of Lightwave Technology*, vol. 38, no. 6, pp. 1125–1137, 2020.

[5] E. Sarikaya and E. Onur, "Placement of 5G RAN slices in multi-tier O-RAN 5G networks with flexible functional splits," in *2021 17th International Conference on Network and Service Management (CNSM)*, 2021, pp. 274–282.

[6] B. Ojaghi, F. Adelantado, and C. Verikoukis, "So-ran: Dynamic ran slicing via joint functional splitting and mec placement," *IEEE Transactions on Vehicular Technology*, pp. 1–16, 2022.

[7] W. da Silva Coelho, A. Benhamiche, N. Perrot, and S. Secci, "Function splitting, isolation, and placement trade-offs in network slicing," *IEEE Transactions on Network and Service Management*, vol. 19, no. 2, pp. 1920–1936, 2022.

[8] O-RAN Alliance, "Cloud architecture and deployment scenarios for O-RAN virtualized RAN," Tech. Rep., Mar. 2022, version 3.00.

[9] PASSION Project, "Definition of use cases and requirements for network, systems and subsystems," https://www.passion-project.eu/wp-content/uploads/2018/10/D2.1_v1.0.pdf, 2018.

[10] X. Liu, "Enabling optical network technologies for 5G and beyond," *Journal of Lightwave Technology*, vol. 40, no. 2, pp. 358–367, 2022.

[11] "IMT vision - framework and overall objectives of the future development of IMT for 2020 and beyond," International Telecommunications Union, Rec. ITU-R M.2083-0, 2015.

[12] "Verticals URLLC use cases and requirements," NGMN Alliance, Frankfurt, Germany, Tech. Rep., Feb. 2020.

[13] "5G for connected industries and automation," 5G Alliance for Connected Industries and Automation, Frankfurt, Germany, Tech. Rep., Feb. 2019.