Dynamic, Reconfigurable and Green Network Slice Admission Control and Resource Allocation in the O-RAN Using Model Predictive Control

Nikolaos Fryganiotis, Eleni Stai, Ioannis Dimolitsas, Anastasios Zafeiropoulos, Symeon Papavassiliou
Institute of Communication and Computer Systems (ICCS)
School of Electrical and Computer Engineering
National Technical University of Athens, Athens, Greece
nfryganiotis@netmode.ntua.gr, estai@mail.ntua.gr, jdimol@netmode.ntua.gr, tzafeir@cn.ntua.gr, papavass@mail.ntua.gr

***Abstract*—In the context of 5G, virtualization has transformed Radio Access Network (RAN) architectures, enabling efficient resource utilization, flexibility and scalability of RAN deployments and operations. Within this paradigm, network slicing has emerged as a pivotal technique, enabling the creation of tailored virtual network instances to meet diverse service requirements. This study performs joint slice request admission control and optimal Virtual Network Functions (VNFs) placement in O-RAN-enabled networks, subject to infrastructure and Quality-of-Service constraints. Contrary to existing schemes, emphasis is placed on two not yet deeply studied directions. The first is about handling future uncertainties on slice requests arrivals for which an iterative Model Predictive Control (MPC)-based approach is proposed that leverages updated traffic forecasts for dynamic adaptation. The second relates to VNFs migration in the O-RAN modules to increase the slice acceptance ratio in an energy efficient way. Through performance evaluations and comparisons, we demonstrate the efficacy of the proposed MPC-based solution compared to other approaches.**

***Index Terms*—Network Slicing, Radio Access Network (RAN), Open RAN (O-RAN), Network Function Virtualization, Admission Control, Model Predictive Control (MPC)**

I. Introduction

In recent times, the telecommunications industry has witnessed a significant evolution driven by the integration of Edge and Cloud computing within the context of fifth-generation (5G) networks. These advancements have fundamentally transformed the management of network resources, introducing novel levels of scalability and operational efficiency. Central to this evolution are virtualization technologies such as Network Function Virtualization (NFV) [1] and Software-Defined Networking (SDN) [2], which enable the abstraction and virtualization of network functions, thereby facilitating dynamic resource allocation and optimization.

A pivotal aspect of this evolution is network slicing, representing a departure from traditional network architectures towards a modular service delivery approach. Network slicing enables the creation of customized virtualized network instances tailored to specific applications and tenants [3]. By isolating network resources into independent slices, each characterized by unique attributes and performance goals, service providers can efficiently deliver a diverse set of services [4]. These services encompass ultra-reliable low-latency communication (URLLC), enhanced mobile broadband (eMBB), and massive machine type communication (mMTC), among others, while maximizing operational efficiency.

This modular approach to service delivery is complemented by the disaggregation of the Radio Access Networks (RAN) architecture. Among the most recognized RAN disaggregation approaches is the O-RAN architecture introduced by the Open RAN (O-RAN) Alliance[1] in order to enable RAN disaggregation, openness and interoperability. This architecture divides the RAN into three key components: the Central Unit (CU), the Distributed Unit (DU), and the Radio Unit (RU), which can be deployed on open hardware and cloud nodes as VNFs.

Network slicing in O-RAN, is intricately linked to the placement of RAN-specific Network Functions (NFs) in the RU, DU, and CU. Such a disaggregated architecture allows leveraging edge computing to enable distributed placement of O-RAN NFs and minimize network delay. By deploying DUs closer to RUs at the network edge, operators can reduce latency and improve the overall performance of RAN slices. However, in this regard, network slicing in O-RAN is mapped into a complex RU, DU, and CU resource allocation problem. Challenges arise in the dynamic allocation of these resources to support varying slice requirements and changing slice request patterns, while minimizing power consumption and reconfiguration costs associated with VNF migration towards improving slice admittance ratio [5]. Generally, VNF allocation is performed either proactively but assuming perfect forecasts of future slice admission requests for a quite long time horizon (e.g., [6]) or reactively upon arrival of the slice requests with future knowledge on traffic arrivals in an expected sense (e.g., [7], [8]).

The above challenges underscore the need for innovative solutions to optimize resource utilization, minimize network delay and power consumption but also importantly enhance the robustness of O-RAN slicing deployments under uncertain-

ISBN 978-3-903176-63-8 © 2024 IFIP

[1] https://www.o-ran.org

ties on future knowledge. Our work contributes significantly towards this direction by solving the problem of optimal joint slice admission control and VNFs placement in the O-RAN modules with an iterative Model Predictive Control (MPC) strategy that allows considering updated forecasts of future slice arrivals. Also, it aims to shed light on the issue of minimizing the reconfiguration costs associated with optimizing multiple slice deployments, which are related to slice downtime (decreased slice availability), offering insights into strategies to streamline this process and ensure maximization of revenue during slice admission. We appropriately handle reconfiguration along the MPC iterations to improve slice admittance in an energy efficient way. Additionally, the proposed setting considers vendors' Quality of Service (QoS) issues such as end-to-end delays, but also, an overall green operation through accounting for power consumption costs.

The remainder of the document is organized as follows. In Section II the state-of-the-art is presented, and the contributions of the current study are highlighted. Section III illustrates the modeling of the adopted O-RAN architecture, its components, and the respective structure of a network slice request. The system dynamics and the discussed problem formulation are presented in Section IV, while the proposed solution is thoroughly described in Section V. The evaluation of the latter is shown in Section VI, followed by the study's conclusions in Section VII.

II. Background & Contributions

There exists several works in the recent literature studying the problem of slice admission control as well as VNFs placement over the disaggregated 5G RAN physical architecture consisting of RUs, DUs and CUs, also referred to as function split. The work in [9] studies the joint problem of optimal function split and slicing in the 5G RAN where slicing extends up to the user level and different service requirements are taken into consideration. Also, [10] studies the joint traffic routing and function split problem in the 5G RAN while handling diverse constraints but focusing on optimizing the level of centralization, without consider power consumption issues as we do in this paper. In addition, the work of [11] formulates a general placement problem with functional split options and solves it as a Binary Integer Linear Programming (BILP) problem in three stages. The objective is to minimize the computing resources cost and maximize the aggregation of radio functions but does not consider delay requirements of the slices as our formulation does. Moreover, the work in [6] considers a very similar setting with our work for placing slices in the RAN, but with the sole goal of energy efficiency. However, in all the above approaches, contrary to our work, a single-shot multi-period optimization problem is solved either assuming knowledge or using forecasts for future traffic arrivals and neither updated forecasts are considered nor slice reconfiguration is performed.

Dynamic resource allocation for RAN is mostly combined with Reinforcement Learning (RL)-based solutions. The paper [7] allocates resource blocks, transmit power, and computational resources to network slices that are requested by the users in a stochastic manner for downlink communications at a 5G base station. The objective is to maximize the weighted sum of satisfied requests over a time horizon, subject to communication and computational constraints and Q-learning is employed for the solution. In [8] online admission and placement of RAN slices in an O-RAN enabled network is considered. The goal is to maximize a two-factor profit that includes the long-term revenue from accepted slices minus an idle cost of servers' deployment, subject to capacity and delay constraints. The solution is based on a Deep Reinforcement Learning (DRL) algorithm, the Proximal Policy Optimization (PPO). Also, in the work of [12], a radio resource slicing problem is examined that tries to share the total bandwidth over the slices. The objective is the maximization of the long-term expectation reward, which tries to balance the costs from resource utilization with the Quality of Experience (QoE) satisfaction ratio. A Deep Q-learning solution is adopted. Again [13] solves a similar problem but with focus on the green operation by considering renewable energy sources that can power the DUs and CUs, without considering extensive service requirements. Finally, [14] focuses on function split for base stations and uses a safe reinforcement learning approach to deal with system constraints based on long short-term memory networks. But in all the existing RL-based works, neither updated forecasts (e.g., different than those used for training), nor VNFs migration are explicitly considered.

MPC-based decision making, as in the proposed approach in this work, is applied for VNF placement but in different settings than RAN. For instance, in the work of [15], the authors propose an MPC-based scheme for jointly performing request admission, resource activation, VNF placement, resource allocation, and traffic routing. However, again neither updated forecasts for slice requests, nor VNFs migrations are explicitly considered. VNF migration is considered in [16] in a different context than the RAN architecture and only for solving a single-shot multi-period optimization problem without performing dynamic reconfiguration with updated forecasts at each iteration as in the current work.

The contributions of this work are summarized as follows:

- We formulate the joint optimization problem of slice requests admission control, VNFs placement, VNFs migration and traffic routing considering the disaggregated O-RAN architecture. The goal is to maximize the long term revenue of slice acceptance minus costs deriving due to re-allocation of VNFs and power consumption. From Mixed Integer Non-linear, we transform the problem formulation to Mixed Integer Linear (MILP).
- Importantly, compared to the literature we additionally formally consider VNFs migration in our model that can become key for maximizing revenues from slice acceptance or minimizing costs from power consumption via aggregating VNFs to utilize the available computing resources as efficiently as possible.
- Also, we provide an MPC-based solution approach to handle uncertainties for future slice requests and exploit updated forecasts for slice request arrivals. We appropri-

ately adapt the optimization problem formulation so that it can be integrated within an MPC solution framework, e.g., already accepted slices from a previous MPC iteration should remain accepted in next MPC iterations. We provide a pseudo-code to detail the MPC iterations and required state and forecast updates.
- We demonstrate the performance improvements of our dynamic VNFs placement and migration scheme that considers updates on future information via comparisons with single-shot optimization solutions as well as with approaches that do not reconfigure slice placement.

III. System Architecture and Modeling

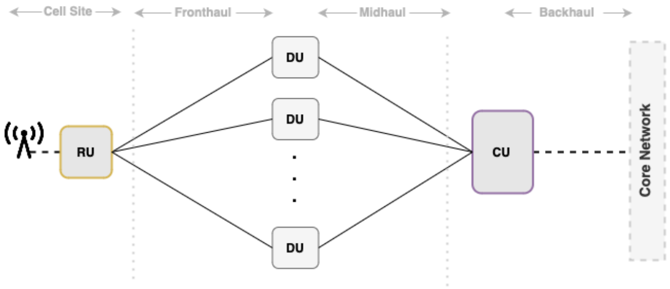


Figure 1: Proposed O-RAN-based Architecture.

A. Substrate Network Model

Figure 1 depicts the deployed O-RAN based architecture. In detail, micro-datacenters, namely Edge Clouds (ECs), are deployed at the network edge and serve as computing resources, in the proximity of the radio unit enabling low-latency processing and reducing backhaul traffic. Let $\mathcal{E}$ denote the set of ECs of the topology. Each edge cloud (EC) $e \in \mathcal{E}$ hosts a DU responsible for processing and managing network functions associated with specific network slices. The ECs are connected with the cell-cite, where an RU is deployed, via fronthaul (FH) connections, while midhaul (MH) links connect each EC with the Regional Cloud (RC) datacenter, denoted by $\mathcal{R}$, where the CU is deployed. The RC serves as a centralized computing resource for higher-level processing and coordination across multiple ECs. The FH links facilitate low-latency communication between the RU and the DUs, while MH links provide high-bandwidth connectivity between DUs and CU. For every $e \in \mathcal{E}$ the total computing capacity, in CPU cores is defined as CE_e, while the corresponding parameter for the regional cloud is denoted by CR. Furthermore, transmission delay of the FH and the MH links is defined as $\delta_{r,e}$, and $\delta_{e,\mathcal{R}}$, $\forall e \in \mathcal{E}$, where r is the RU. Moreover, $CB_{F,e}$, $CB_{M,e}$ stand for the bandwidth of the FH and MH links associated with the EC $e \in \mathcal{E}$, respectively.

B. Slice Request Model

In the proposed O-RAN-based system modeling, we consider a set F consisting of available VNFs, denoted by $v_f \in F$, that can be deployed to compose various network slices. It is important to note that certain VNFs, specifically the VNFs with IDs v_0, v_1, remain consistent across all network slice requests. In precise, VNFs v_0, v_1, are the initial VNFs used in every request s. When a network slice request s arrives it is composed of a subset of F, represented by F_s. Thus, for every network slice $F_s = \{v_0^s, v_1^s, \dots, v_f^s, \dots, v_{n_s}^s\} \subseteq F$, it stands that $v_0^s = v_0 \in F$, and $v_1^s = v_1 \in F$. Notably, each network slice request s is structured following the Service Function Chain (SFC) deployment model, where a specific execution sequence is defined [17]. This sequence dictates the order in which the VNFs $v_f \in F_s$ are processed within the network slice. In detail, the initial VNF v_0^s and subsequent VNFs v_f^s, for $f = 1, \dots, n_s$ are arranged according to the SFC model, ensuring that the network slice functions as intended and meets the requirements specified by the request. Additionally, the compute and network related resource requirements are defined per slice s. For a VNF $v_f \in F_s$ there are specific demands regarding CPU cores for the VNF deployment $c_{s,f}$ and the bandwidth for the link $(f-1, f)$ denoted as $b_{s,f}$. Furthermore, each request s arrives at a specific time t_s and has a holding time ht_s, indicating the duration for which the slice remains active once requested. Moreover, an end-to-end delay requirement $D_{max,s}$ and a priority value pr_s is defined for each slice request, reflecting its tolerance level for delay and importance, respectively.

IV. Problem Formulation

In this section, we formulate optimal VNFs allocation problem that maximizes slice acceptance while minimizing costs of re-allocating accepted slices and power consumption.

A. Notation Table & Assumptions

Table I collects the variables and parameters used in our problem formulation. The placement of VNFs is described by time dependent binary variables $x_{s,f}^e(t)$, $y_{s,f}(t), \forall e \in \mathcal{E}, f \in F_s, s \in R(t)$, which indicate if they are placed on an EC, $e \in \mathcal{E}$, or the RC, correspondingly. In particular:

$$x_{s,f}^e(t) = \begin{cases} 1, & f \in F_s \text{ is placed on EC } e \in \mathcal{E} \text{ at t}, \\ 0, & \text{otherwise.} \end{cases}$$

$$y_{s,f}(t) = \begin{cases} 1, & f \in F_s \text{ is placed on RC at t}, \\ 0, & \text{otherwise.} \end{cases}$$

Also, given the binary $X_s(t)$ that denotes if a slice request has been admitted at time t or earlier (Table I), we further define a variable $sr_s(t)$ indicating if a slice s is active (i.e., admitted and not expired) at a particular time t.

$$sr_s(t) = \begin{cases} X_s(t), & t_s \leq t < (t_s + ht_s), \\ 0, & \text{otherwise.} \end{cases}$$

Note that we consider that each EC hosts a single distributed computing unit. The VNF with index 0 is placed on the RU, and the remaining VNFs are placed either on an EC or the RC with the constraint that if a VNF is placed on an EC all its preceding VNFs in the path should be placed on ECs.

Notation	Description
H	Time horizon of the MPC
$\mathcal{E}$	Set of ECs
EC	Edge Cloud
RC	Regional Cloud
RU	Radio Unit
F	A set of available VNFs
F_s	A subset of VNFs consisting a service chain for slice s
$f \in F_s$	Index of a VNF in the service chain of slice s
F_s^{-l}	VNFs associated with slice s except the VNF with index l
n_s	Number of VNFs of slice s
t_s	Arrival time of slice s
ht_s	Holding time of slice s
pr_s	Priority value of slice s
$R(t)$	Set of newly arrived, waiting and already active slices at time t
$R^{fixed}(t)$	set of already admitted slices that are still active at the MPC-iteration starting at t
$\tilde{R}(\ell)$	set of existing active slices and slice requests that have arrived but not yet admitted with positive updated holding time
$c_{s,f}$	CPU requirement of VNF f of slice s
$b_{s,f}$	Bandwidth requirements for two successive VNFs $f-1, f$ of slice s
$D_{max,s}$	End-to-end delay requirement of slice s
$X_s(t)$	1 if slice s has been admitted at time t or earlier otherwise 0
$x_{s,f}^e(t)$, $y_{s,f}(t)$	Binaries indicating the placement of VNFs
$sr_s(t)$	Binary indicating if a slice is active
$C_e(t)$	CPU utilization on cloud $e \in \mathcal{E}$
$B_e(t)$	Bandwidth utilization between the RU and the EC $e \in \mathcal{E}$
CE_e	Total possible computing capacity of EC e for every time t
CR	Total possible computing capacity of the RC
$CB_{F,e}$ $(CB_{M,e})$	Total bandwidth of FH (MH) link related to e
ξ_{E-R}	Cost for moving a VNF from an EC to the RC
ξ_{E-E}	Cost for moving a VNF between ECs
$u_e(t)$	1 if the fronthaul link connected to EC e is utilized
$v_e(t)$	1 if the midhaul link connected to EC e is utilized
$\delta_{r,e}$, $\delta_{e,\mathcal{R}}$	Delay parameters
P^{max}	Maximum power consumption of an EC
γ	Proportion of the consumed power of an idle server with respect to P_{max}
P_{net}^{max}	Maximum power consumption of network links
$P_{net,e}^{fix}$	Fixed power consumption of a network link between RU and $e \in \mathcal{E}$
$XX_{s,f}^e(t)$, $XY_{s,f,f+1}^e(t)$, $XY_{s,f}^e(t)$, $YX_{s,f}^e(t)$	Auxiliary variables for the linearization of the problem formulation

Table I: Selective Notation and Description.

B. System Dynamics

Here, the dynamic equations that model the evolution of the CPU, and bandwidth utilization of the ECs are described. The CPU utilization of $e \in \mathcal{E}$ evolves as follows:

$$C_e(t+\Delta\tau) = C_e(t) + C_e^{ALL}(t+\Delta\tau) - C_e^{REL}(t+\Delta\tau), \quad (1)$$

where $C_e^{REL}(t)$ and $C_e^{ALL}(t)$ stand for the released and newly allocated, respectively, CPU resources on EC e at time t.

To express mathematically $C_e^{REL}(t)$, both expiring slices as well as reconfigurations of active slices are considered. In particular, a slice that arrived at time t_s with holding time ht_s, at time $t_s + ht_s$ expires and releases all its allocated resources. Furthermore, the current resources of admitted slices may be released followed by a re-allocation of new resources. As a result, $\forall e \in \mathcal{E}$,

$$C_e^{REL}(t+\Delta\tau) = \sum_{s \in R(t+\Delta\tau)} \sum_{f \in F_s} x_{s,f}^e(t)(1 - x_{s,f}^e(t+\Delta\tau))c_{s,f}. \quad (2)$$

To express mathematically $C_e^{ALL}(t)$, both new slices that get admitted at time t are considered, and VNFs are placed on EC $e \in \mathcal{E}$ as well as already active slices that are reconfigured and receive resources on EC e. Thus, $C_e^{ALL}(t)$ is equal to:

$$C_e^{ALL}(t+\Delta\tau) = \sum_{s \in R(t+\Delta\tau)} \sum_{f \in F_s} (1 - x_{s,f}^e(t))x_{s,f}^e(t+\Delta\tau)c_{s,f}. \quad (3)$$

If calculating the quantity $C_e^{ALL}(t+\Delta\tau) - C_e^{REL}(t+\Delta\tau)$ that appears in (1), the quadratic terms cancel out. As a result:

$$C_e(t+\Delta\tau) = C_e(t) + \sum_{s \in R(t+\Delta\tau)} \sum_{f \in F_s} (x_{s,f}^e(t+\Delta\tau) - x_{s,f}^e(t))c_{s,f}. \quad (4)$$

We now proceed in defining the dynamic equation determining the evolution of the bandwidth utilization over the FH links, i.e., those between the RU and the ECs. In detail:

$$B_e(t+\Delta\tau) = B_e(t) + B_e^{ALL}(t+\Delta\tau) - B_e^{REL}(t+\Delta\tau), \ \forall e \in \mathcal{E}, \quad (5)$$

where $B_j^{REL}(t)$ and $B_j^{ALL}(t)$ stand for the released and newly allocated, respectively, bandwidth between the RU and the EC e at time t. Specifically, in order to model the allocation and release of bandwidth for links connecting the RU and an EC we should only consider the second VNF of the service chain of each slice (i.e., the VNF with index 1). Therefore,

$$B_e^{REL}(t+\Delta\tau) = \sum_{s \in R(t+\Delta\tau)} x_{s,1}^e(t)(1 - x_{s,1}^e(t+\Delta\tau))b_{s,1}, \quad (6)$$

$$B_e^{ALL}(t+\Delta\tau) = \sum_{s \in R(t+\Delta\tau)} (1 - x_{s,1}^e(t))x_{s,1}^e(t+\Delta\tau)b_{s,1}. \quad (7)$$

Again, the quadratic terms are eliminated, and the final equation for the evolution of the bandwidth utilization over the FH links $\forall e \in \mathcal{E}$ is given by:

$$B_e(t+\Delta\tau) = B_e(t) + \sum_{s \in R(t+\Delta\tau)} (x_{s,1}^e(t+\Delta\tau) - x_{s,1}^e(t))b_{s,1}. \quad (8)$$

C. Constraints

In this part, we define the instantaneous (i.e., for every time t) system constraints under which optimal network slicing should be performed. To begin with, the aggregate of the computing resources to be binded in any EC or the RC has to be lower than the total possible computing capacity of the corresponding cloud, i.e.,

$$\sum_{s \in R(t)} \sum_{f \in F_s} x_{s,f}^e(t)c_{s,f} \leq CE_e, \ \forall e \in \mathcal{E}, \quad (9)$$

$$\sum_{s \in R(t)} \sum_{f \in F_s} y_{s,f}(t)c_{s,f} \leq CR. \quad (10)$$

Next, there exists bandwidth constraints over transmission links. Note that a slice uses a link between the RU and an EC, if its second VNF (i.e., with index 1) is placed in this EC. The bandwidth constraints for the fronthaul and midhaul links are expressed for every time t as follows.

$$\sum_{s\in R(t)} x^e_{s,1}(t) b_{s,1} \le CB_{F,e},\ \forall e \in \mathcal{E}, \tag{11}$$

$$\sum_{s\in R(t)} \sum_{f\in F_s^{-n_s}} x^e_{s,f}(t) y_{s,f+1}(t) b_{s,f+1} \le CB_{M,e},\ \forall e \in \mathcal{E}. \tag{12}$$

For every admitted slice s, its first VNF (i.e., with index 0) is placed in the RU. Consequently, it cannot be placed in either an EC or the RC, i.e.,

$$\sum_{\forall e\in\mathcal{E}} x^e_{s,0}(t) = 0,\ \forall s \in R(t), \tag{13}$$

$$y_{s,0}(t) = 0,\ \forall s \in R(t). \tag{14}$$

In addition, a VNF f, of an admitted slice s, can be allocated either to a single EC or the RC at every time t, i.e.,

$$\sum_{e\in\mathcal{E}} x^e_{s,f}(t) + y_{s,f}(t) = sr_s(t),\ \forall s \in R(t), \forall f \in F_s^{-0}. \tag{15}$$

Moreover, for an admitted slice, the VNF 1 should be placed in an EC, which is guaranteed if it cannot be placed in the RC, i.e.,

$$y_{s,1}(t) = 0,\ \forall s \in R(t). \tag{16}$$

Under the assumptions of service chaining and colocation, if for an admitted slice s, a VNF f is placed in an EC, the VNFs preceding f in the service chain, i.e., $1, \ldots, f-1$, should be also placed in the same EC. Similarly, if a VNF is placed in the RC, its successive VNFs in the service chain of the slice should be also placed in the RC. Therefore,

$$x^e_{s,f}(t) \le x^e_{s,f-1}(t),\ \forall f \in F_s^{-0,1}, \forall s \in R(t), \forall e \in \mathcal{E}, \tag{17}$$

$$y_{s,f}(t) \le y_{s,f+1}(t),\ \forall f \in F_s^{-n_s}, \forall s \in R(t). \tag{18}$$

The total delay imposed by FH and MH links should not exceed the $D_{max,s}$ requirement of a slice s at any time t.

$$\sum_{e\in\mathcal{E}} x^e_{s,1}(t)\delta_{r,e} + \sum_{f\in F_s^{-n_s}} \sum_{e\in\mathcal{E}} x^e_{s,f}(t) y_{s,f+1}(t) \delta_{e,R} \le D_{\max,s},\ \forall s \in R(t). \tag{19}$$

A FH link is considered utilized only if one or more slices have placed their VNFs with index 1 in its corresponding EC, i.e.,

$$u_e(t) \ge x^e_{s,1}(t),\ \forall s \in R(t), \forall e \in \mathcal{E}, \tag{20}$$

$$u_e(t) \in \{0,1\},\ \forall e \in \mathcal{E}. \tag{21}$$

A MH link is considered utilized if for any pair of two successive VNFs of any slice, one is placed in the EC and the other on the RC, i.e.,

$$v_e(t) \ge x^e_{s,f}(t) y_{s,f+1}(t),\ \forall s \in R(t), \forall f \in F_s^{-n_s}, \forall e \in \mathcal{E}, \tag{22}$$

$$v_e(t) \in \{0,1\},\ \forall e \in \mathcal{E}. \tag{23}$$

Finally, a slice that gets admitted at time t should be considered admitted for its entire control lifecycle, i.e.,

$$X_s(t+\Delta\tau) \ge X_s(t),\ \forall s \in R(t). \tag{24}$$

D. Objective Function

To define the objective function we consider three factors, namely: (i) the revenue obtained from slice acceptance, (ii) the cost deriving from reallocating already accepted slices, and (iii) the power consumption of the ECs and the network links that are utilized for the slice deployment. The revenue of a slice acceptance at time t is

$$ReV(t) = \sum_{s\in R(t)} sr_s(t) \cdot pr_s. \tag{25}$$

For the reallocation cost both VNFs moving from an EC to the RC or vice versa and those VNFs that move from an EC to another are considered. The instantaneous reallocation cost of VNFs from an EC to the RC or vice versa is expressed as:

$$ReC^{E-R}(t) = \sum_{s\in R(t), f\in F_s, e\in\mathcal{E}} \left(x^e_{s,f}(t) y_{s,f}(t+\Delta\tau) + y_{s,f}(t) x^e_{s,f}(t+\Delta\tau)\right)\xi_{E-R}, \tag{26}$$

while the reallocation cost at t from an EC to a different EC is expressed by

$$ReC^{E-E}(t) = \sum_{s\in R(t), f\in F_s, e\in\mathcal{E}, i\in\mathcal{E}-e} x^e_{s,f}(t) x^i_{s,f}(t+\Delta\tau)\xi_{E-E}. \tag{27}$$

Regarding the power consumption cost, we follow the modelling of a power efficient VNF placement approach from the literature [18]. In case of ECs, it is given at t by:

$$PC^{EC}(t) = \sum_{e\in\mathcal{E}} \left(u_e(t)\gamma P^{max} + (1-\gamma)\frac{C_e(t)}{CE_e} P^{max}\right). \tag{28}$$

In case of links that connect the RU with an EC e, it is formulated as

$$PC^{RU-E}(t) = \sum_{e\in\mathcal{E}} \left(u_e(t) P^{fix}_{net,e} + \frac{B_e(t)}{CB_{F,e}} P^{max}_{net}\right). \tag{29}$$

Finally, in case of links that connect an EC e with the RC, it can be written as

$$PC^{E-R}(t) = \sum_{e\in\mathcal{E}} v_e(t) P^{fix}_{net,e}. \tag{30}$$

E. Optimization Problem

The optimization problem to be solved is formulated as:

Problem 1.

$$\begin{aligned} \max \sum_{t=\ell:\Delta\tau:\ell+(H-1)\Delta\tau} &\Big(ReV(t) - ReC^{E-R}(t) - ReC^{E-E}(t) \\ &- PC^{EC}(t) - PC^{RU-E}(t) - PC^{E-R}(t)\Big) \cdot \Delta\tau \\ \text{s.t}\quad & (4), (8), (9)-(22), \\ & X_s(t) \in \{0,1\},\ x^e_{s,f}(t), y_{s,f}(t) \in \{0,1\}, \\ & \forall s \in R(t), \forall f \in F_s, \forall e \in \mathcal{E}, \forall t \in \{\ell, \ell+\Delta\tau .., \ell+(H-1)\Delta\tau\}, \end{aligned} \tag{31}$$

where $C_e(\ell), B_e(\ell), \forall e \in \mathcal{E}$ are given. The optimization problem is mixed integer quadratically constrained problem with quadratic objective. Next we apply Watters' linearization [19] on the quadratic terms in both the objective function and the constraints and the problem takes a MILP form.

F. Linearization of Quadratic Terms

The quadratic terms of the Problem (31) are the following: $x^e_{s,f}(t)x^i_{s,f}(t+\Delta\tau)$, $x^e_{s,f}(t)y_{s,f+1}(t)$, $x^e_{s,f}(t)y_{s,f}(t+\Delta\tau)$ and $y_{s,f}(t)x^e_{s,f}(t+\Delta\tau)$. To employ the Watters' linearization, we introduce the following auxiliary variables $XX^{e,i}_{s,f}(t+\Delta\tau)$, $XY^e_{s,f,f+1}(t)$, $XY^e_{s,f}(t+\Delta\tau)$, $YX^e_{s,f}(t+\Delta\tau)$ that will respectively replace the above quadratic terms. In addition, we add constraints for the auxiliary variables to the problem formulation. Indicatively, for the variables $XX^{e,i}_{s,f}(t)$, these are formulated as follows $\forall s \in R(t), \forall f \in F_s, \forall e \in \mathcal{E}$:

$$XX^{e,i}_{s,f}(t+\Delta\tau) \geq x^e_{s,f}(t) + x^e_{s,f}(t+\Delta\tau) - 1, \tag{32}$$

$$XX^{e,i}_{s,f}(t+\Delta\tau) \leq x^e_{s,f}(t) + x^i_{s,f}(t+\Delta\tau), \tag{33}$$

$$XX^{e,i}_{s,f}(t+\Delta\tau) \in \{0,1\}. \tag{34}$$

Corresponding constraints should be added in Problem (31) for the remaining auxiliary variables.

V. Proposed Solution - Model Predictive Control

To perform dynamic optimal slice admission and resource allocation on admitted slices, we will solve the Problem 1 in a Model Predictive Control (MPC) fashion (Figure 2). Algorithm 1 presents a pseudo-code of the solution process. The control period starts at t_0 where no slices have arrived yet and thus no computing and bandwidth resources have been yet allocated. Problem 1 is then solved with initial time t_0 and a horizon of H time intervals in the future each of duration $\Delta\tau$. The number of slices and their arrival times within the future time horizon H is unknown and forecasts are used. In this work, forecasts are considered given by an external forecasting tool. The decisions about slice admittance and resource allocation are obtained for all time intervals within the horizon H. However we apply only the decisions for time t_0 and disregard all other decisions for future times. By the time we apply the decisions we also observe which slices actually arrived. For slices that were forecasted to arrive but did not, we cancel any related resource allocation decision. For slices that arrived without being expected, we also do not allocate resources as otherwise infeasibilities and high costs may emerge.

At the next decision time, i.e., $t_0 + \Delta\tau$, the process is repeated. In particular, we observe the updated states regarding the computing resources of the ECs and the RC as well as the bandwidth of the links. Also, updated forecasts of the number and arrival times of new slices are obtained for a time horizon equal again to H time intervals in the future each of duration $\Delta\tau$. However, slices that have been already accepted at time t_0 or earlier, should continue providing service at time $\ell = t_0 + \Delta\tau$, if their updated holding time is positive.

This is because if a slice gets admitted it should be considered admitted for its entire lifecycle. Note that for active slices we reduce their holding time by $\Delta\tau$ from an MPC iteration to another. This requirement cannot be directly handled by Problem 1 and necessitates an additional constraint that is given below:

$$X_s(\ell) = 1, \forall s \in R^{Fixed}(\ell), \tag{35}$$

with $R^{fixed}(\ell)$ the set including all slices satisfying $s \in \tilde{R}(\ell-$

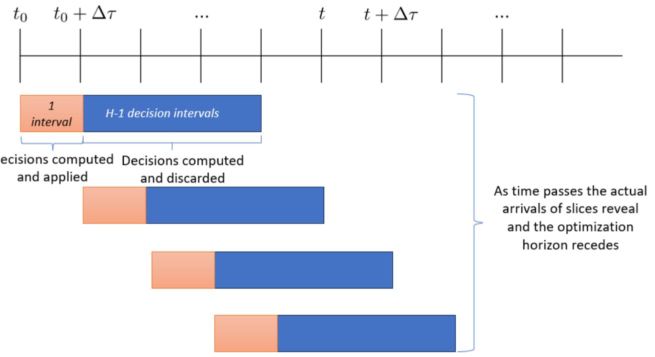

t_0
$t_0 + \Delta\tau$
...
t
$t + \Delta\tau$
...
1 interval
H-1 decision intervals
Decisions computed and applied
Decisions computed and discarded
As time passes the actual arrivals of slices reveal and the optimization horizon recedes

Figure 2: Illustration of MPC receding horizon.

Algorithm 1 Model Predictive Control for Dynamic Resource Allocation on Network Slices

Input: Time horizon H; Initial time t_0; All parameters related to the RU, EC, RC, including power consumption, bandwidth, computing availability, re-allocation cost parameters;

```
procedure MPC
Initializations:
    $\ell \leftarrow t_0$; $C_e(t_0) \leftarrow 0$; $B_e(t_0) \leftarrow 0, \forall e \in \mathcal{E}$;
    while not the end of control period do
        1. Observe the state variables, $C_e(\ell)$, $B_e(\ell), \forall e \in \mathcal{E}$.
        2. If $\ell > t_0$, observe already active slices with positive updated holding time, $ht_s \leftarrow ht_s - \Delta\tau$, forming the set $R^{Fixed}(\ell)$. Also, compute the binary parameters $x^{e,Fixed}_{s,f}$, $y^{Fixed}_{s,f}$, for all slices $s \in R^{Fixed}(\ell)$.
        3. Receive updated forecasts of arriving slices for a horizon $H$, i.e., for times $\{\ell, \ell+\Delta\tau, ..., \ell+(H-1)\Delta\tau\}$;
        4. Solve Problem 2 and obtain the main optimization variables, i.e., $X_s(\ell)$, $x^e_{s,f}(\ell)$, $y_{s,f}(\ell)$ for all $s \in R(\ell)$, all related VNFs $f$ and all times $\{\ell, \ell+\Delta\tau, ..., +(H-1)\Delta\tau\}$;
        5. Observe the realizations of the uncertain quantities at the current decision interval $\ell$, i.e., $\tilde{R}(\ell)$;
        6. Keep the decisions only for time $\ell$, i.e., $X_s(\ell)$, $x^e_{s,f}(\ell)$, $y_{s,f}(\ell)$ for all $s \in \tilde{R}(\ell)$ (and related VNFs) and discard those of all future time slots, i.e., $\{\ell+\Delta\tau, ..., \ell+(H-1)\Delta\tau\}$. If a slice $s$ was forecasted to arrive but in reality did not or the vice versa set $X_s(\ell) = x^e_{s,f}(\ell) = y_{s,f}(\ell) = 0, \forall e \in \mathcal{E}, f \in F_s$.
        7. Update the state variables $C_e(\ell)$, $B_e(\ell)$ for the next decision interval $(\ell + \Delta\tau)$ using the equations of Section IV-B for slices in $\tilde{R}(\ell)$.
        8. $\ell \leftarrow \ell + \Delta\tau$
    end while
end procedure
```

$\Delta\tau$) and $ht_s \geq \Delta\tau$ and $X_s(\ell - \Delta\tau) = 1$. Thus, the variable $X_s(\ell)$ should be set to 1 at the current decision interval t for all slices that have been accepted in a previous MPC round and their holding time is still positive in the current round. At each time ℓ, $\tilde{R}(\ell)$ includes existing active slices, slice requests that have arrived but not yet admitted with positive updated holding time as well as newly-arrived slices (in reality, i.e., not as predicted by forecasts). In addition, the Problem 1 should be adapted in order to account for the potential re-allocation costs of slices between times t_0 and $t_0 + \Delta\tau$ (or generally between times $\ell - \Delta\tau$ and ℓ).

To do so we introduce new binary parameters $x_{s,f}^{e,Fixed}, y_{s,f}^{Fixed}$, for all slices $s \in R^{Fixed}(\ell)$ with values set as $x_{s,f}^{e,Fixed} = x_{s,f}^{e}(\ell - \Delta\tau)$, $y_{s,f}^{Fixed} = y_{s,f}(\ell - \Delta\tau)$. Based on the above, we formulate Problem 2 that is an adapted version of Problem 1 for being integrated in an MPC framework.

Problem 2.

$$\begin{aligned}
\max \sum_{t=\ell:\Delta\tau:\ell+(H-1)\Delta\tau} \Big(& ReV(t) - ReC^{E-R}(t) - ReC^{E-E}(t) \\
& - PC^{EC}(t) - PC^{RU-E}(t) - PC^{E-R}(t) \Big) \cdot \Delta\tau \\
& - REC^{E-R,Init}(\ell) \cdot \mathbf{1}_{\ell > t_0} - REC^{E-E,Init}(\ell) \cdot \mathbf{1}_{\ell > t_0} \\
\text{s.t. } & (4), (8), (9) - (22), \\
& X_s(t) \in \{0,1\},\ x_{s,f}^{e}(t), y_{s,f}(t) \in \{0,1\}, \\
& \forall s \in R(t), \forall f \in F_s, \forall e \in \mathcal{E}, \forall t \in \{\ell, \ell + \Delta\tau .., \ell + (H-1)\Delta\tau\}, \\
& \text{if } \ell > t_0 \text{ include Equation (35),}
\end{aligned} \tag{36}$$

where,

$$\mathbf{1}_{\ell > t_0} = \begin{cases} 1, \text{if } \ \ell > t_0 \\ 0, \quad \text{otherwise,} \end{cases} \tag{37}$$

$$ReC^{E-R,Init}(\ell) = \sum_{s \in R^{Fixed}(\ell), f, e} \big(x_{s,f}^{e,Fixed} y_{s,f}(\ell) + y_{s,f}^{Fixed} x_{s,f}^{e}(\ell) \big) \xi_{E-R} \cdot \Delta\tau, \tag{38}$$

$$ReC^{E-E,Init}(\ell) = \sum_{s \in R^{Fixed}(\ell), f} \sum_{e \in \mathcal{E}} \sum_{i \in \mathcal{E}^{-e}} x_{s,f}^{e,Fixed} x_{s,f}^{i}(\ell) \cdot \Delta\tau. \tag{39}$$

The same process is repeated for all times within the control period that might be larger than H. Figure 2 illustrates the receding horizon of MPC as well as for which intervals decisions are computed and applied or computed and discarded.

VI. Evaluation Results & Comparisons

This section presents the assessment of the proposed methodology designed to address Problem 1, through modeling and simulations. For the implementation of the simulation environment, version 3.10 of Python programming language is used. We follow an object oriented programming approach, defining one class for the slice request model and one class for implementing the solution methods, i.e., the proposed mpc-based method and the alternative solutions. The substrate network parameters are involved in the solution class. To solve the optimization problem we use Gurobi solver, specifically, the gurobipy Python package. The substrate network parameters are mentioned in Table II, while the slice parameters at Table III. The substrate network consists of three ECs and one RC. We consider two types of slices, URLLC and emBB. VNF requirements adhere to a typical paradigm commonly for cloud service providers. These requirements manifest in three distinct flavors, denoted as *small*, *medium*, and *large*. Each flavor corresponds to varying levels of resource demands, particularly in terms of CPU cores for our modeling. Specifically, the CPU demand per flavor is specified as 2 cores for *small*, 4 for *medium*, and 8 for *large*. In the context of the simulation process, a flavor is chosen equiprobably for each VNF of every slice. We consider that the number of slice requests is equal to 15 over a time horizon of 12 time units. In order to perform a fair evaluation between the distinct approaches, we assume that the holding time of every deployed slice could not exceed the 12 timesteps setup which reflect to 24 hours of deployment time. To ensure that the holding time of a slice after its admission does not exceed the remaining evaluation period, variable $\zeta_s = 13 - t_s$ is defined as an upper bound for the holding time of the slice request s.

Parameter	Value
Number of ECs	3
EC, RC capacity	16, 64 cores
FH link capacity, delay	2Gbps, 4ms
MH link capacity, delay	4Gbps, 8ms
γ	0.8
P^{max}	2000W
P_{net}^{max}	200 W
P_{net}^{fix}	160 W

Table II: Substrate network parameters.

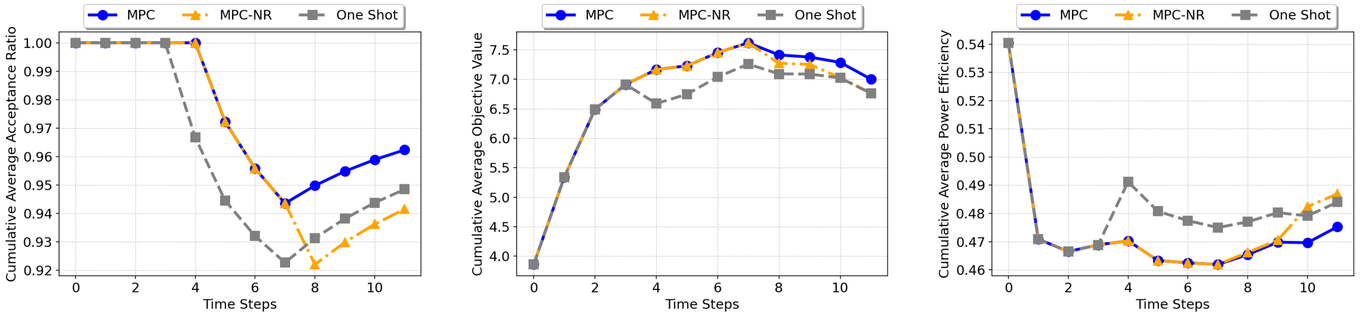
Attributes	Slice type	
	URLLC	eMBB
$D_{max,s}$	25 ms	50 ms
$c_{s,f}$	$\in \{2,4,8\}$	$\in \{2,4,8\}$
$b_{s,f}$	100 Mbps	200 Mbps
Request arrival times t_s	$\mathcal{U}\{1,12\}$	$\mathcal{U}\{1,12\}$
Holding time ht_s	$\min(\mathcal{U}\{3,6\}, \zeta_s)$	$\min(\mathcal{U}\{3,6\}, \zeta_s)$
Number of requests	55% of total	45% of total
Normalized priority	3 per time-slot	2.4 per time-slot

Table III: Slice Request Parameters.

We generate forecasts for the time arrival of requests using the following forecasting method. Initially, the arrival time of requests is sampled from a discrete uniform distribution over the optimization horizon, H. At each time slot of the control period, we solve the Problem 1 and obtain the decision variables. We consider that our forecasting method generates forecasts that are inaccurate with probability 10%. In this context, we define two forecasting scenarios.

Less accurate forecasting scenario: In this scenario, the arrival time of not yet realized slice requests is resampled from a discrete uniform distribution over the horizon.

Highly accurate forecasting scenario: Under the highly accurate forecasting scenario, 20% of the expected requests to arrive resample their time arrival.

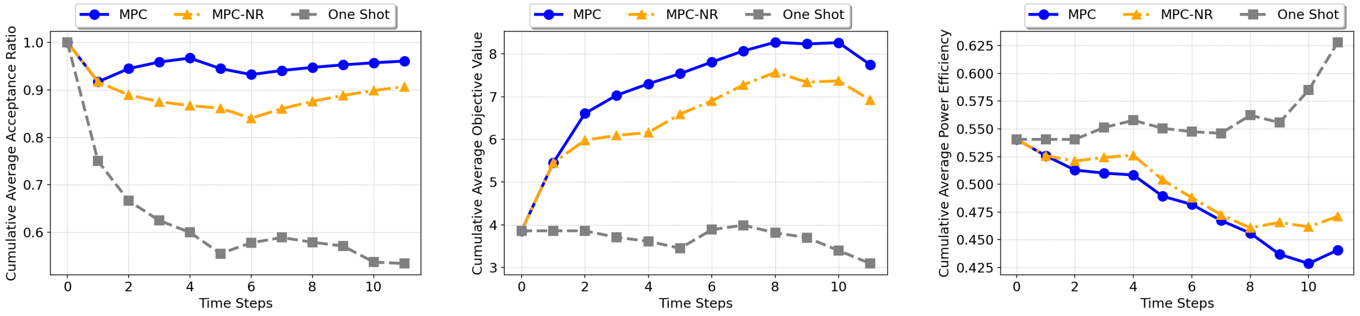


(a) Cumulative average acceptance ratio.

(b) Cumulative average objective value.

(c) Cumulative average power efficiency.

Figure 3: Comparative evaluation results under highly accurate forecasting scenarios.



(a) Cumulative average acceptance ratio.

(b) Cumulative average objective value.

(c) Cumulative average power efficiency.

Figure 4: Comparative evaluation results under less accurate forecasting scenarios.

All the simulations are executed in an Ubuntu 20.04 virtual machine with 8 vcpus and 8GB of RAM of an Intel(R) Xeon(R) CPU@2.10GHz server.

A. *Assessed Methods and Evaluation Metrics*

The evaluation focuses on comparing the performance of three distinct methods: the proposed MPC solution, denoted as MPC, an MPC variant that avoids VNF reallocation ($MPC-NR$), and an one-shot optimization approach that decides the admission of slice requests at the first time slot for the entire horizon ($One\ shot$). These methods were tested under the two different settings of slice request forecasting that were discussed above, in order to assess their robustness and adaptability to dynamic changing of slice request demands. The evaluation metrics for the performance assessment are:

- **Acceptance Ratio:** The acceptance ratio measures the percentage of admitted slices by a certain time step, determined by the active slice subset, which includes ongoing requests yet to expire. It reflects the system's efficacy in handling incoming slice demands amidst existing deployment commitments.
- **Objective Value:** This metric represents the optimization objective value achieved by each method based on the actual realization of slice requests, offering insights into their efficiency in resource allocation and utilization during the slice requests admission.
- **Power Efficiency:** This is defined as the ratio of revenue generated by the admitted slices over the total power consumption of the compute and network counterparts of the substrate network. The inverse of power efficiency signifies the system's effectiveness in conserving energy, with lower values denoting higher power efficiency.

It is worth mentioning that normalization of objective metrics is performed prior to simulations to mitigate disparities arising from the diverse scales of objective-related values.

B. *Results and Discussion*

Here, we present the results of simulations comparing the proposed Model Predictive Control (MPC) approach alongside two alternative optimization strategies, MPC-NR and One Shot, as these discussed above. Figure 3 and Figure 4 present the cumulative average of the above evaluation metrics computed over a 12-time step horizon for the two cases of the forecasting scenarios, aiming to provide a comprehensive overview of the performance trends observed across the simulation.

In scenarios with favorable forecast conditions, marginal differences are observed between the solution methods. However, upon closer examination, MPC demonstrates its adaptability over the prediction horizon, particularly in achieving higher acceptance ratios, as shown in Figure 3a. At the same time, it maintains optimal values for other key metrics compared to MPC-NR and One Shot solutions (Figures 3b, 3c), showcasing its ability to adjust resource allocation decisions regarding VNF placement, while achieving to maintain amlow consumption power of the compute and network counterparts.

The efficacy of the MPC approach becomes more evident in less accurate forecast scenarios. The evaluation results regarding this scenario are shown in Figure 4. In more detail, despite inherent uncertainties, MPC consistently outperforms One Shot optimization method, highlighting its robustness and resilience to forecast inaccuracies. Moreover, compared to the MPC solution that totally eliminates the reallocation of VNFs, namely the MPC-NR, the proposed MPC approach maintains a substantial performance advantage across all evaluated metrics More precisely, the optimal resource utilization is highlighted in Figure 4a, where the cumulative average of acceptance ratio is much higher than the other approaches from very early during the evaluation period and maintained for the whole horizon, as reflected in Figure 4b. It is worth mentioning, that despite the higher acceptance ratio, which entails to increased resource demand, the proposed MPC approach still outperforms the MPC-NR and One Shot methods in terms of power efficiency (Figure 4c).

The observed performance disparities underscore the significance of proactive and adaptive resource allocation strategies in dynamic network environments. While traditional optimization methods may suffice under ideal conditions, the inherent uncertainty of real-world scenarios necessitates more sophisticated approaches. The MPC ability to leverage forecast information to anticipate demand fluctuations and proactively optimize resource allocation decisions is a key determinant of its efficacy on slice admission in O-RAN-based architectures.

Furthermore, the performance advantage of the MPC-based approach over MPC-NR reveals the importance of considering reallocation in dynamic resource allocation strategies. By factoring in these costs, the MPC framework effectively manages the trade-offs between resource usage optimization and the operational overhead associated with reallocating and migrating VNFs. This ensures optimal resource utilization while ensuring higher slice availability with minimal management complexities from the infrastructure provider's perspective.

VII. Conclusions

In this work, we tackled the problem of joint slice admission control and resource allocation in the O-RAN architecture while considering uncertainties in the slice request arrival process and allowing for the possibility of VNF reallocation towards increasing slice acceptance subject to a green operation. Contrary to the existing literature that often assumes accurate knowledge on future slice arrivals, our proposed approach is based on MPC that allows dynamically adapting out decisions to the emerging system conditions and updated forecasts. We have performed a thorough evaluation, showcasing that the proposed MPC-based scheme outperforms state-of-the art static approaches that do not include VNF imigration possibilities by boosting the revenue from slice acceptance in a power efficient way. In future work, we plan to integrate machine learning approaches for ensuring the applicability of the proposed dynamic scheme in fast time scales of control.

Acknowledgment

This research work was supported partly by the European Commission through the Horizon Europe/JU SNS project 6Green (Grant Agreement no. 101096925).

References

[1] "ETSI GS NFV 006 V2.1.1, Architectural Framework Specification." https://www.etsi.org/deliver/etsi_gs/nfv/001_099/006/02.01.01_60/gs_nfv006v020101p.pdf, 2021. (accessed on 10 February 2024).

[2] J. Ordonez-Lucena, P. Ameigeiras, D. Lopez, J. J. Ramos-Munoz, J. Lorca, and J. Folgueira, "Network Slicing for 5G with SDN/NFV: Concepts, Architectures, and Challenges," *IEEE Communications Magazine*, vol. 55, no. 5, pp. 80–87, 2017.

[3] R. V. Rosa and C. E. Rothenberg, "The Pandora of Network Slicing: A Multicriteria Analysis," *Transactions on Emerging Telecommunications Technologies*, vol. 31, no. 1, p. e3651, 2020.

[4] G. Tseliou, F. Adelantado, and C. Verikoukis, "Netslic: Base Station Agnostic Framework for Network Slicing," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 3820–3832, 2019.

[5] E. Amiri, N. Wang, *et al.*, "Deep Reinforcement Learning for Robust VNF Reconfigurations in O-RAN," *IEEE Transactions on Network and Service Management*, 2023.

[6] N. Sen and A. F. A, "Towards Energy Efficient Functional Split and Baseband Function Placement for 5G RAN," in *IEEE 9th International Conference on Network Softwarization (NetSoft)*, pp. 237–241, 2023.

[7] Y. Shi, Y. E. Sagduyu, and T. Erpek, "Reinforcement Learning for Dynamic Resource Optimization in 5G Radio Access Network Slicing," in *IEEE 25th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, pp. 1–6, 2020.

[8] N. Sen and A. F. A, "Intelligent Admission and Placement of O-RAN Slices Using Deep Reinforcement Learning," in *IEEE 8th International Conference on Network Softwarization (NetSoft)*, pp. 307–311, 2022.

[9] B. Ojaghi, F. Adelantado, *et al.*, "Sliced-RAN: Joint Slicing and Functional Split in Future 5G Radio Access Networks," in *IEEE International Conference on Communications (ICC)*, pp. 1–6, 2019.

[10] C. C. Erazo-Agredo, M. Garza-Fabre, *et al.*, "Joint Route Selection and Split Level Management for 5G C-RAN," *IEEE Transactions on Network and Service Management*, vol. 18, no. 4, pp. 4616–4638, 2021.

[11] F. Z. Morais, G. M. F. de Almeida, *et al.*, "PlaceRAN: Optimal Placement of Virtualized Network Functions in Beyond 5G Radio Access Networks," *IEEE Transactions on Mobile Computing*, vol. 22, no. 9, pp. 5434–5448, 2023.

[12] R. Li, Z. Zhao, *et al.*, "Deep Reinforcement Learning for Resource Management in Network Slicing," *IEEE Access*, vol. 6, pp. 74429–74441, 2018.

[13] T. Pamuklu, M. Erol-Kantarci, and C. Ersoy, "Reinforcement Learning Based Dynamic Function Splitting in Disaggregated Green Open RANs," in *IEEE International Conference on Communications (ICC)*, pp. 1–6, 2021.

[14] F. W. Murti, S. Ali, and M. Latva-Aho, "Constrained Deep Reinforcement Based Functional Split Optimization in Virtualized RANs," *IEEE Transactions on Wireless Communications*, vol. 21, no. 11, pp. 9850–9864, 2022.

[15] M. Golkarifard, C. F. Chiasserini, F. Malandrino, and A. Movaghar, "Dynamic VNF Placement, Resource Allocation and Traffic Routing in 5G," *Computer Networks*, vol. 188, p. 107830, 2021.

[16] V. Eramo, E. Miucci, *et al.*, "An Approach for Service Function Chain Routing and Virtual Function Network Instance Migration in Network Function Virtualization Architectures," *IEEE/ACM Transactions on Networking*, vol. 25, no. 4, pp. 2008–2025, 2017.

[17] I. Dimolitsas, D. Dechouniotis, and S. Papavassiliou, "Time-efficient Distributed Virtual Network Embedding for Round-trip Delay Minimization," *Journal of Network and Computer Applications*, p. 103691, 2023.

[18] A. Varasteh, B. Madiwalar, *et al.*, "Holu: Power-aware and delay-constrained vnf placement and chaining," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1524–1539, 2021.

[19] L. J. Watters, "Reduction of integer polynomial programming problems to zero-one linear programming problems," *Operations Research*, vol. 15, no. 6, pp. 1171–1174, 1967.