INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2024, VOL. 70, NO. 4, PP. 953–959
Manuscript received September 18, 2024; revised October, 2024. doi: 10.24425/ijet.2024.152082

The joint orchestration of edge applications and UPF CNFs over edge-cloud continuum infrastructure in 6G

Witold Jóźwiak, Andrzej Bęben, and Maciej Sosnowski

***Abstract*—The paper focuses on future 6G mobile systems deployed over the edge-cloud continuum infrastructure. The challenge is designing an effective orchestration method that allocates instances of edge applications and user plane functions, addressing the diverse requirements of involved stakeholders. We propose, implement, and evaluate new joint orchestration algorithms that take advantage of the abstract representation of edge-cloud continuum resources. The evaluation based on mathematical MILP models and trials in an experimental edge-cloud continuum environment confirmed that the proposed joint orchestration outperforms other approaches.**

***Keywords*—orchestration; beyond 5G; 6G networks; edge-cloud continuum; MILP; experiments**

I. Introduction

FUTURE mobile networks, such as beyond 5G (B5G) and 6G systems, evolve towards completely software-defined solutions. The challenge is designing a new architecture where the Radio Access Network (RAN) and the Core Network (CN) functions are designed as Cloud-native Network Functions (CNFs) ready for deployment in a cloud infrastructure [1], [2]. The "cloudification" of the telecommunication infrastructure makes it much more: i) *flexible*, allowing fast reconfiguration, upgrade and easy development of new network CNFs required by B5G/6G system, ii) *scalable*, mainly by vertical scaling mechanisms adjusting CNFs performance accordingly to the current traffic demands, iii) *effective* due to resource sharing that improves its long term utilization, as well as, iv) *open* enabling seamless integration with edge applications and services offered by third-party providers, and application migration to the network edges improving quality of delay-sensitive applications. Moreover, deploying B5G/6G over the public or private clouds significantly reduces CAPEX costs, which fosters the deployment of private 5G/6G networks.

The expected benefits of 6G systems [3], [4] motivate the research on new challenges that, among others, are: i) the design of edge-cloud continuum (ECC) infrastructure [5] that uniformly represents heterogeneous cloud resources coming from multiple providers such as public cloud providers, edge computing providers, telecommunication operators, etc., ii) addressing the performance concerns of software-defined CNF, primarily related to User Plane Functions (UPF), iii) design of effective orchestration methods that allocates edge applications and CNFs instances matching together their requirements, users' expectations, resource constraints, and diverse policies of providers involved in the service chain.

This paper focuses on the orchestration problem in 6G systems deployed over the ECC infrastructure. We aim to design an efficient orchestration method for the multi-provider system, where instances of edge applications/services and 6G network functions are deployed over the same ECC infrastructure composed of heterogeneous far edge, edge, regional, primary data centers, and public clouds. We analyze three strategies differing in the scope of provider integration. We start from the independent subsystems, designed and orchestrated autonomously, and go through limited cooperation between stakeholders up to the joint strategy, where the orchestrator manages the edge applications/services 6G network functions over the integrated ECC resources. We argue that such a joint orchestration strategy is the most efficient. However, it may be difficult because it needs the close cooperation of stakeholders.

We evaluate the proposed orchestration strategies in the designed model of ECC infrastructure. We use the mixed-integer linear programming (MILP) technique to implement proposed orchestration algorithms. This model allowed us to get the grand true results, deeply understand the system behavior, and explain the relationships between involved stakeholders. Moreover, we set up an experimental ECC environment based on four Kubernetes (K8s) clusters [6], where we deployed a private 5G network using the free5GC [7], UERANSIM software [8], and physical gNB station designed by AMARISOFT. The experiments prove the feasibility of the joint orchestration and confirm its effectiveness over other strategies.

The paper organization is the following: Section II discusses the orchestration problem in the ECC environment, analyses the related work, and gives motivation for our approach. In Section III, we present considered orchestration strategies, model the ECC infrastructure, and formulate joint orchestration of edge applications and CNFs functions as a MILP problem. Then, we present the proposed orchestration algorithms. Their effectiveness is evaluated in Section IV.

This research was partially supported by the Ministry of Science and Higher Education, Poland, grant number KPOD.01.18-IW.03-0009/24, on "National Laboratory for Advanced 5G Research" financed by the European Union NextGenerationEU under National Recovery Plan.

W. Jóźwiak, A. Bęben, and M. Sosnowski are with Institute of Telecommunication, Warsaw University of Technology, Warsaw, Poland (e-mail: {witold.jozwiak, andrzej.beben, maciej.sosnowski}@pw.edu.pl).

Then, in Section V, we present experiments that prove the approach's feasibility. Finally, Section VI summarizes the paper and outlines future works.

II. Problem Statement & State of the Art

Contrary to the currently exploited LTE and 5G networks, the future 6G systems will be deployed over ECC infrastructure. The main benefits of the ECC come from the abstract, uniform representation of heterogeneous resources shared by multiple stakeholders. This model enables the deployment of many subsystems offering diverse services using the same unified distributed cloud-native infrastructure. Each system deploys its orchestrator that manages and controls provided services according to their specific needs. Consequently, the ECC approach leads to better resource utilization, improved elasticity, and scalability and finally benefits users by fostering the cooperation of service providers.

This paper analyzes the case where a 6G network operator and an independent third-party edge application provider deploy their services over the ECC infrastructure. It merges computing resources shared by different providers. The telecommunication operators lease resources at the far/near edges and their local, regional, and primary data centers. The edge providers, e.g., emerging MEC (*Multi-access Edge Computing*) or CDNs (*Content Delivery Network*) providers, deploy computing servers at the network edges or IXPs (*the Internet eXchange Points*). Finally, large cloud service providers, such as Google, AWS, and Microsoft, bring a vital volume of computing resources from their data centers. Participation in the ECC community allows them to provide services closer to the users.

Let us consider an exemplary ECC system presented in Fig. 1, where two service providers are deployed over the ECC infrastructure. The former one, denoted in blue in Fig. 1, is the cloud-native 5G network, while the latter one, denoted in red, corresponds to the MEC service provider as an example of edge/cloud provider. The orchestrators of both systems manage alone the life cycle of offered services. However, as we show later, joint orchestration could significantly improve the effectiveness. The cooperation eliminates the risk of inadequate allocation of microservices of the service chain provided to a user that engages the 5G network to deliver service offered by the cloud providers. Joint orchestration requires a special orchestration algorithm designed to schedule and automatically execute actions related to telco CNFs and edge applications/services, taking into account diverse providers' and user requirements,

We argue that special attention must be paid to properly place UPF instances because they handle PDU *Protocol Data Units* sessions transferring data packets to the user's terminals. In the considered system, we assume that the data transfer path always consists of two types of UPF: i) I-UPF (*Intermediate UPF*) with ULCL (*Uplink Classifier*) functionality – deployed by the operator at the base station or other traffic aggregation points. The operator provisions the I-UPFs following the radio resources available on gNB; ii) A-UPF (*PDU Session Anchor UPF*) terminating PDU sessions and providing connection to other systems, e.g,. to the Internet or other service providers.

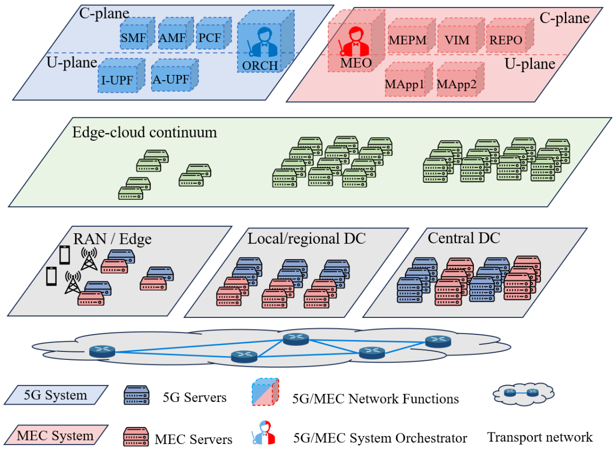

Fig. 1. Exemplary use case with cloud-native 5G network and MEC service provider deployed over ECC infrastructure

The problem of UPF allocation in 5G networks has been widely studied. The authors [9] consider the allocation of UPF service chains distinguishing between UPF types (A-, I-, MI-UPF). The proposed approach optimizes the routing of PDU sessions considering the number and location of UPF instances and the delay experienced on the established paths. The [10] addresses a similar problem where authors consider energy consumption and time-varying traffic conditions. They aim to minimize the total costs by jointly optimizing UPF placement and traffic distribution. Another problem of UPF performance scaling is addressed in [11]. The authors proposed a new scaling algorithm that derives the required number of UPF instances based on queuing models and traffic predictions performed by machine learning algorithms.

On the other hand, the orchestration of edge applications in the edge/fog/MEC environment is a well-studied topic. Many papers, e.g., [12], focus on the challenges of orchestrating services in edge and fog environments, considering technologies like NFV, SDN, and containerization for optimized resource management and scheduling in considered environments.

The joint orchestration of edge applications and UPF in the ECC 6G environment is an emerging topic. The initial considerations and assumptions are discussed in [13], where effective orchestration design is considered as one of the challenges. Our studies fill the gap in the service chain orchestration.

III. Proposed Orchestration Method

This section presents the proposed orchestration strategies and derived model of the ECC infrastructure.

A. Orchestration strategies

We define proposed orchestration strategies based on the stakeholders' ability to cooperate. In particular, we distinguish the 6G telco operator and other third-party cloud providers. The considered strategies are the following:

- S#1: Each provider independently orchestrates its microservices within owned computing resources – A-UPF instances are provisioned and deployed by the telco operator based on the predicted traffic demands while the service provider orchestrates services. However, the telco operator has no prior knowledge about the actual users' demands and used services because of a lack of cooperation between orchestrators;

- S#2 Joint orchestration within owned computing resources – the joint orchestrator makes decisions on both the allocation of service instances and the number and localization of A-UPF instances, taking into account that A-UPF instances would be exclusively deployed on telco resources while applications are exclusively deployed on cloud provider resources;
- S#3 Joint orchestration based on shared computing resources – CNFs can be allocated on any available resources. This strategy assumes close cooperation between orchestrators with complete knowledge of the system's state.

The first strategy represents an approach in which independent orchestrators manage the systems, while others represent joint approaches (*joint orchestration*), where the orchestrators assume close cooperation between providers. The strategy S#1 assumes that the telco operator places A-UPF instances in a network core and at the network edges as a result of network provisioning, The S#2, called joint orchestration, independent computing resources - the orchestrator, in addition to allocating edge application/services, makes decisions on the deployment and number of A-UPF instances, Finally, in S#3 joint orchestration, the orchestrator can deploy and scale A-UPF function instances in shared computing resources. It was also assumed that edge and 5G computing resources could be shared.

B. Model of ECC infrastructure

The proposed orchestration strategies are defined as mixed-integer linear programming (MILP) problems. They are implemented in the MiniZinc language [14] and solved using the HiGHS solver in version 1.5.1 [15].

ECC topology

The ECC infrastructure is modeled as a directed graph $G(A,V,E)$, where A, V represents the set of access and computational nodes, respectively, while E represents the set of connections between computational nodes. Each access node $a_l \in A,\ l=1,\dots,|A|$ is connected to the computational node $v_m \in V,\ m=1,\dots,|V|$. The node assignment is defined by the matrix $P_{A,Q}$, where each element of the matrix $p_{a,q} = [v_m, dist]$ is a vector describing the connection of a given access node to the computational node v_m. Each connection is characterized by its distance $dist$. For example, the matrix element $q_1 = [2, 10]$ means that access node 1 is connected to computational node 2, and the distance between these nodes is 10. The connections between computational nodes $v1, v2 \in V$ are defined by matrix $D_{V,V}$ containing the characteristics of connections $e_j \in E,\ j=1,\dots,|E|$.

In the considered orchestration problem, the reserved resources belong to the two-element set $R = \{CPU,\ BW\}$, which describes the number of computing cores and the throughput of the outgoing virtual link, which takes into account the efficiency of the data transfer layer implementation in the node. The considered problem belongs to the family of multi-criteria problems as long as there are at least two elements in set R. The resources of computing nodes are described by the matrix $H_{V,R}$. The model assumes that the unit cost of a given resource $r \in R$ depends on the location of the computing node, so we model unit costs by the matrix $U_{V,R}$.

CNF descriptor

The instances of edge applications and services and the 6G network functions are launched in the same ECC infrastructure. Therefore, we now refer to both as CNFs (*Cloud Native Functions*). In the CNF set, denoted as K, we distinguish CNFs representing edge applications/services and the A-UPF functions. Due to their different roles in the system, a subset $K^- = K \setminus \{\text{A-UPF}\}$ of the set K is defined, which contains only edge applications/services. Each CNF function is described by the matrices $\Pi_{K,R}$ and M_K defining, respectively, the amount of resources required to launch a single instance of a given CNF and the propagation delay tolerated by the application. In addition, we denote the service capacity offered by a single CNF instance by the M_K matrix. The value $\mu_k \in M_K$ defines the maximum volume of traffic that a single instance can handle, satisfying the requested quality of service level. Additionally, the model uses a matrix $S_{V,K}$ with binary values, which determines whether the node $v_m \in V,\ m=1,\dots,|V|$ is compatible with a CNF of a given type $k_n \in K,\ n=1,\dots,|K|$. This matrix determines which CNFs can run on nodes/resources provided by a particular stakeholder.

Demands arrival

In the modeled system, user requests arrive for a given edge application or service to the access nodes $a_l \in A,\ l=1,\dots,|A|$ at a given time epoch. The number of data streams associated with the application $k_n \in K^-,\ n=\ 1,\dots,|K^-|$ and serviced by access node a_l is described by the matrix $L_{a,k}$.

Decision variables

The orchestration algorithm determines i) the number of instances of a given CNF type that should run on each node and ii) the traffic distribution between the running edge application/service instances towards the access node going through the path of A-UPF instances. These values contain two decision variables describing the state of the system. The variable x_v^k describes the number of CNF instances of type $k_n \in K$, $n=1,\dots,|K|$ that should be launched on the computational node $v_m \in V,\ m=1,\dots,|V|$. The variable $y_{v,u}^{a,k}$ describes the traffic flow, i.e., it determines what volume of the traffic arriving at the access node $a_l \in A,\ l=1,\dots,|A|$ concerning application k, is serviced by the edge application/service instance running on the computation node v_m and the A-UPF instance running on the node $u_m \in V,\ m=1,\dots,|V|$.

Constraints

The equation (1) ensures that the traffic directed to a given node cannot exceed the capacity of all edge application/service instances running on it.

$$\sum_{a\in A}\sum_{u\in V} y_{v,u}^{a,\ k} \ \le\ x_v^k \ \cdot\ \mu_k;\ \forall\ k\ \in K^-,\ \ \forall\ v\ \in V \qquad (1)$$

In the case of A-UPF, a separate constraint is defined because this function handles traffic from all edge applications/services running on a given node. Constraint (2) ensures that the traffic volume from edge applications/services handled by A-UPF functions running on a given node cannot exceed the capacity of the A-UPF functions allocated to that node.

$$\sum_{a \in A} \sum_{k \in K^-} \sum_{v \in V} y_{v,u}^{a,k} \leq x_u^{a-upf} \cdot \mu_{a-upf}; \quad \forall u \in V \tag{2}$$

The equation (3) ensures that running CNF cannot exceed the compute resources of used nodes. It means that the sum of the resources of CNF instances running on a given node cannot exceed its capacity.

$$\sum_{k \in K} x_v^k \cdot \pi_{k,r} \leq H_{v,r}; \quad \forall v \in V, \quad \forall r \in R \tag{3}$$

The equation (4) forces the running of CNF instances only on their designated nodes.

$$x_v^k > 0 \Leftrightarrow S_{v,k} = 1; \; \forall k \in K, \quad \forall v \in V \tag{4}$$

The equation (5) ensures that the delay value tolerated by a given edge application/service is satisfied. The delay value is calculated as the path length, expressed in km, divided by the propagation rate factor $C_v \left[\frac{km}{ms}\right]$. We consider only the propagation delay in the calculation. Packet processing delays are neglected. The condition (5) is checked only for relations with allocated flows.

$$\begin{gathered} C_v \cdot (P_{a,Dist} + D_{P_{a,Node},u} + D_{v,u}) \leq \tau_k \\ \forall a \in A, \; \forall k \in K^-, \; \forall v \in V, \; \forall u \in V; \; y_{v,u}^{a,\;k} > 0 \end{gathered} \tag{5}$$

Objective function

The main goal of the orchestration algorithm is to minimize two factors - the cost of launching the CNF instances and the cost associated with data transmission on the path: access node, I-UPF, A-UPF, edge application/service instances.

The cost of running a CNF instance of type k on node v is determined as the product of the volume of resources r used by instance k and the unit cost specific to a given node v. By multiplying this product by the number of instances (variable x) and summing over all resource types, we obtain the cost of allocating all k instances on node v. The total cost of resource allocation is described by the function f_1.

$$f_1() = \sum_{v \in V} \sum_{k \in K} \sum_{r \in R} x_v^k \cdot \rho_{k,r} \cdot U_{v,r} \tag{6}$$

When calculating the transmission costs, the algorithm considers the sum of the distances on the path between the access node, I-UPF, A-UPF, and the computational node where the application instance is running. The I-UPF function is assumed to be placed in the computational node where the access node is connected.

The costs related to radio transmission are neglected because they do not influence the allocation. In order to ensure unit consistency, the parameter $C_t \left[\frac{1}{Mbit \cdot km}\right]$ was introduced, understood as the cost of transmitting 1 Mb of data over a distance of 1 km. The total transmission cost is defined by the function f_2.

$$\begin{aligned} f_2() = \; & C_t \cdot \sum_{a \in A} \sum_{k \in K^-} \sum_{v \in V} \sum_{u \in V} y_{v,u}^{a,k} \\ & \cdot \pi_{k,BW} \cdot (P_{a,Dist} + D_{P_{a,Node},u} + D_{u,v}) \end{aligned} \tag{7}$$

In B5G/6G systems, great attention is paid to energy conservation. By minimizing the number of nodes running CNF instances, we minimize the system's operating costs. A server that does not have any CNFs can remain in low-power mode and wait for allocation.

The objective function includes the total cost associated with running the computational nodes described by the f_3 function. The C_o parameter is introduced, which is the unit cost of running a computational node.

$$f_3() = C_o \cdot \sum_{v \in V} F\left(\sum_{k \in K} x_v^k\right)$$
$$where \; F(z) = \begin{cases} 0 \; for \; z \leq 0 \\ 1 \; for \; z > 0 \end{cases} \tag{8}$$

The objective function also includes a component related to incomplete satisfaction of requests (rejected requests) described by the function f_4. The degree of incomplete satisfaction of requests is expressed as the difference between the volume of allocated traffic and requests. Considering this criterion requires relaxing the constraint (1), since this constraint forces all requests to be satisfied. The volume of unsatisfied requests is multiplied by the coefficient C_r, which is a penalty for rejection. The coefficient must be sufficiently large so that the algorithm does not prefer rejection.

$$f_4() = C_r \cdot \sum_{\substack{a \in A \\ k \in K^-}} \left[\max\left(0, L_{a,k} - \sum_{\substack{v \in V \\ u \in V}} y_{v,u}^{a,k}\right)\right] \tag{9}$$

The formalization of the algorithm leads to a multi-criteria problem with possibly contradicting objectives (minimizing transmission costs means placing resources closer to the edge of the network, which increases allocation costs). A weighted sum of criteria was used to transform the model into a single-criteria one. Assuming that the model includes Ω criteria, weights γ were introduced such that:

$$\sum_i \gamma_i = 1; \; i = 1, \; 2, \; \ldots, \Omega \tag{10}$$

Normalizing the sum of weights and appropriately selecting their values allows us to influence the optimization process and provide a trade-off between criteria. The objective function may be defined as a linear combination of particular criteria. These conditions are formally written as follows:

$$\begin{aligned} & minimize \; f() = \sum_i \gamma_i \cdot f_i(); \; i = 1, \; 2, \ldots, \Omega \\ & subject \; to \; : \; (1) - \; (5) \end{aligned} \tag{11}$$

IV. Performance evaluation

Our experiments aim to evaluate the effectiveness of the presented orchestration strategies. Our studies were performed under high system load conditions at the limit of loss occurrence. Losses, i.e., failure to satisfy a fraction of requests, can occur due to failure to meet latency requirements, lack

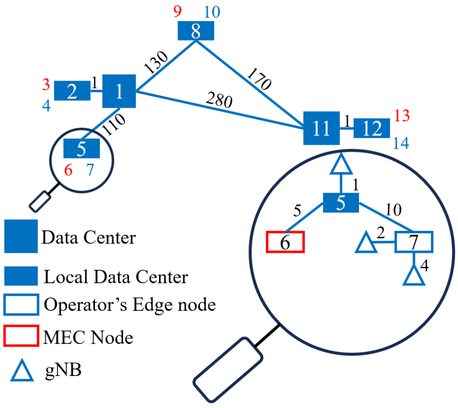

Fig. 2. Topology of the evaluated system

of network or computing resources, or overload of A-UPF instances.

The tests were carried out in the system shown in Fig. 2, consisting of 14 compute nodes ($v_1 - v_{14}$) and 12 access nodes ($a_1 - a_{12}$), representing a small or medium-sized network operator. There are 4 types of computing nodes in the system: (1) central data center, (2) local data center, (3) operator edge node, and (4) the edge computing node. To each node of type (2), there is attached one access node (gNB), and to each node (3) are attached 2 gNBs each. The topology assumes the existence of operator data centers in Warsaw and Poznan. It is assumed that nodes (1) each have 10000 units of resources and cost 1 cost unit, nodes (2) each have 1,000 units of resources and cost 5 units, and nodes (3) and (4) each have 100 units of resources and cost 10 units. In typical deployments, these costs are the lowest in large data centers due to the high resource aggregation and increase the closer the computing servers are to the network edge.

TABLE I
DESCRIPTORS OF THE CNFS

CNF	Required resources $\Pi_{K,R}$		Tolerable latency	Handling capacity	Arrival of requests $\forall\ a \in A$
	CPU	BW	τ_k $[ms]$	μ_k $[req]$	$L_{a,k} [req]$
k_1	10	10	-	100	-
k_2	6	6	1	1	5
k_3	2	4	10	100	100

There are 3 types of microservices running on the system, with significantly different requirements: the A-UPF function is represented as application k_1, edge app1 (k_2) is a demanding application that requires a lot of resources and tolerates only low latency of 1 ms, and edge App2 (k_3) characterize moderate requirements. The application descriptors are summarized in Table I.

It was assumed that all access nodes received 5 requests each for k_2 and 100 requests for k_3, for a total of 1260 requests. This value allowed the system to enter a high load state. This data was also used to determine the required number of A-UPF (13 instances) in the S#1 strategy. All experiments were performed in PL5G lab research infrastructure [16]. The experiments used an HPE ProLiant DL380 server with the following specifications: Intel(R) Xeon(R) CPU E5-2650 v3 @ 2.30GHz, 40 cores, RAM 128 GB, Ubuntu 20.04.6 LTS (kernel version 5.4.0-146-generic). Computations were performed using the MiniZinc integrated development tool version 2.7.6 [14] together with the HiGHS [15] solver in version 1.5.1. HiGHS was chosen for the evaluation because it was the fastest among other solvers available in MiniZinc.

TABLE II
COMPARISON OF THE EFFICIENCY OF ORCHESTRATION STRATEGIES

Value	*S#1*	*S#2*	*S#3*	*max S#3*
f_1	8 580	9 120	5 080	90 880
f_2	229 076	184 812	16 123	45 758 328
f_3	10 000	10 000	8 000	14 000
f_4	200 000	200 000	0	0
ψ [%]	0.16	0.16	0.0	0.0
$Obj.\ function$	111 914	100 983	7 301	11 465 802

The experiment results are shown in Tab. II and III. Analyzing the results, it was found that for the S#1 and S#2 strategies, requests for k_3 applications were rejected due to lack of resources. In addition, the independent orchestration assumed in the S#1 strategy leads to higher transmission costs due to the mismatched location of the A-UPF relative to the allocated applications. Allowing the orchestrator to adjust the number and location of A-UPF instances under S#2 reduced this effect at the expense of running more A-UPFs. Note that S#2, with the same traffic, allocated 3 more A-UPF instances than S#1 (16 vs. 13). The additional A-UPF instances reduced the transmission cost, resulting in a lower overall cost.

The mismatch is also indicated by the occurrence of losses. Implementing the S#3 joint orchestration strategy makes it possible to reduce resource usage and eliminate losses. As a result, the S#3 strategy increases the handled traffic. The load limit for S#3 is 63 k_2 application requests per node and 12,000 k_3 application requests. The results for this case are provided in the last column of Tab. II and III. S#1-3 strategies can support applications that require very low latency, whereby: the constraint for S#1 is the operator's correct estimation of A-UPF capacity at the network edge, and for S#2 the constraint is resources available at the network edge.

Due to its characteristics, S#1 is not resistant to A-UPF overload – the lack of scaling of these functions leads to requests being rejected. S#2-3 strategies launch additional A-UPF instances when overloaded.

TABLE III
PLACEMENT OF CNF INSTANCES IN NODES

W	*S#1*			*S#2*			*S#3*			*max load S#3*		
#	k_1	k_2	k_3	k_1	k_2	k_3	k_1	k_2	k_3	k_1	k_2	k_3
v_1	3	0	0	1	0	0	1	14	1	637	187	627
v_2	2	0	0	4	0	0	0	0	0	69	0	77
v_3	0	14	4	0	14	4	0	0	0	6	2	7
v_4	0	0	0	0	0	0	3	1	2	7	0	7
v_5	2	0	0	3	0	0	2	15	1	1	165	0
v_6	0	15	2	0	15	2	0	0	0	0	16	1
v_7	0	0	0	0	0	0	2	0	2	4	8	3
v_8	2	0	0	3	0	0	2	15	1	1	165	0
v_9	0	15	2	0	15	2	0	0	0	0	16	1
v_{10}	0	0	0	0	0	0	2	0	2	4	8	3
v_{11}	2	0	0	1	0	0	2	15	1	657	173	598
v_{12}	2	0	0	4	0	0	0	0	0	55	0	112
v_{13}	0	14	4	0	14	4	0	0	0	0	16	0
v_{14}	0	0	0	0	0	0	2	0	2	7	0	7

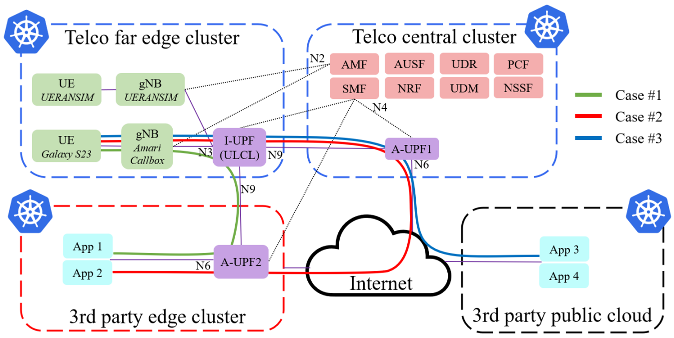

Fig. 3. The experimental ECC environment

For any optimization, the size of the decision space is an important aspect. The time required to determine the solution by S#1-2 strategies increases non-linearly, but despite this, optimal results are obtained in an acceptable time. For S#3, the times needed to determine the solution increase significantly for larger topologies and workloads.

V. Experiments

The promising results of the evaluation presented in section IV motivate us to verify the joint approaches in a practical ECC environment. We aim to prove their feasibility and identify any obstacles if they occur during deployment. In our experiments, we use cloud-native 5G implementation deployed over the exemplary ECC environment comprising four Kubernetes (K8s) clusters as presented in Fig. 3.

It covers a far-edge telco cluster running RAN#1, implemented based on UERANSIM emulation [8], and the I-UPF with ULCL (uplink classifier) functionality. This UPF also serves the physical RAN#2 created by Amarisoft gNB and several Galaxy S23 terminals. We deployed 5G Core in the second cluster using the Free 5GC [7] software. We adopt the helm charts from the Towards5GS-helm [17] project to deploy 5GC. This cluster also includes the instance of A-UPF that terminates PDU sessions and provides the default connectivity to the Internet. Moreover, we set up the edge computing cluster dedicated to edge applications and services third-party providers offer. The joint approach will also use this cluster to deploy the edge A-UPF instance and handle local traffic. Finally, the fourth cluster represents any cloud providers available on the Internet, such as AWS, GCP, Azure, etc.

The discussed clusters are deployed in four locations, connected to the core network differently, and managed by independent stakeholders. The telco far edge clusters are co-located with the RAN network and connected to the core network, where the 5GC cluster is also connected. Edge and cloud clusters are connected to the Internet or the Internet Exchange Points (IXPs), or if an appropriate agreement between telco and edge provider exists, they are connected to the edge network. We introduce some impairments to model propagation delays of the mentioned earlier connections. By default, Kubernetes assumes that pods have only one network interface. It was necessary to add more network interfaces to enable a correct implementation of the 5G core in a containerized form. The Multus CNI network plugin was used for this purpose. All clusters were interconnected through a transport network, providing connectivity between network functions. The addressing for the network functions was designed so that a separate IP subnet was allocated for each 5G interface, ensuring isolation in the address space. In addition, two networks were created to act as data networks (DNs) on the N6 interface for the edge cluster and the far cloud. All clusters were set up based on Kubernetes v1.23, with both master and worker nodes running Ubuntu 20.04 LTS with kernel version 5.4.0-146-generic (gtp5g module compliant). Free5GC v3.1.1 was used.

In this experiment, we consider the deployment of an exemplary Augmented Reality (AR) application, which is CPU-demanding and delay-sensitive. We apply orchestration strategies presented in section III and analyze the application performance under the following test cases:

- Case #1: The joint orchestration algorithm deploys the application and A-UPF instances at the edge cloud. This case will prove the feasibility of the joint approach and show the expected benefits over the other test cases.
- Case #2: The orchestrator decides to deploy the application at the edge cloud. However, the A-UPF at the edge is overloaded, so the traffic will be handled by the default A-UPF located in the 5G core network.
- Case #3: The orchestrator cannot allocate the application at the edge, so it is deployed in the cloud provider infrastructure.

In all considered cases, we deploy the AR application and then measure the packet transfer characteristics in the uplink direction, i.e., the packets were generated at UE and sent toward the instance of the edge application. We model packet flow as the UDP stream with packets of fixed size (1250B) sent at 100, 200, and 300 packets per second, corresponding to low, medium, and high loads, respectively. We measure packet transfer characteristics expressed by IPTD (*IP Packet Transfer Delay*), IPDV (*IP Delay Variation*) with 99 percentile, and IPLR (*IP Packet Loss Rate*). In each test case, we send over 10,000 packets, and we repeat measurements three times for different traffic loads to get credible results. The obtained results are presented in the Fig. 4.

The experiment confirms that the joint orchestration of UPF functions and edge applications/services is feasible. The instance of A-UPF can be deployed outside the telco cluster. However, secure and trust communication between pods must be assured to connect A-UPF to the I-UPF located at the telco edge and the SMF (*Session Management Function*) deployed inside the 5G core. Moreover, the result indicates that the edge application can be deployed at the edge if, and only if, the edge A-UPF is available and properly provisioned. Such a situation occurs in case #1, so we observe that the packet transfer delay is about a few ms. Otherwise, if A-UPF is unavailable at the edge, the edge application should be allocated nearest to the default A-UPF location. So, running the edge application in a centralized cloud is even better than at the edge, as packet transfer delays in case #3 are lower compared to case #2. Note that the orchestration of edge applications and services independently from the orchestration of A-UPF

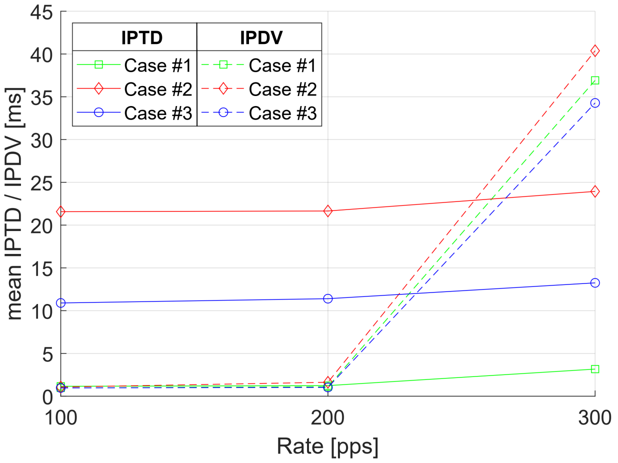

Fig. 4. The packet transfer characteristic

is ineffective and may not only eliminate the gain coming from edge computing but may also degrade offered services. We argue that the proposed joint orchestration approaches eliminate the discussed drawbacks, enable the effective use of ECC resources, and may also improve the quality of offered services.

VI. Summary

The paper deals with future 6G mobile systems deployed over the edge-cloud continuum infrastructure. We proposed and evaluated the efficient orchestration strategies for the multi-provider system, where instances of edge applications and cloudified 6G network functions are deployed over the ECC infrastructure composed of heterogeneous far edge, edge, regional, and large centralized data centers. We developed the MILP model to evaluate the proposed orchestration strategies. The obtained results confirmed that joint orchestration outperforms, particularly under high traffic loads, where resource constraints and service latency become critical. Moreover, we set up an experimental ECC environment and carried out trials to validate the feasibility and effectiveness of our proposed orchestration methods. We observed that the joint deployment of edge applications and UPF instances led to especially in high-load conditions.

Future research will focus on further enhancements of the joint orchestration, focusing on the horizontal scaling process of UPF instances based on real-time traffic predictions. We also analyze the application of machine learning techniques for traffic forecasting in resource orchestration. Future work would also explore the applicability of the joint orchestration strategies to low-latency applications such as augmented/mixed reality, industrial IoT, and autonomous driving to assess their scalability and effectiveness in various use cases.

References

[1] O. Bulakcı, X. Li, M. Gramaglia, A. Gavras, M. Uusitalo, P. Rugeland, and M. Boldi, *Towards sustainable and trustworthy 6G: Challenges, enablers, and architectural design*. Now Publishers, 2023. [Online]. Available: https://doi.org/10.1561/9781638282396

[2] T. Chen, S. Kuklinski, E. Pateromichelakis, K. Samdanis, A. Kourtis, N. Nikaein, and A. Skarmeta, "Service-oriented architecture evolution towards 6g networks," in *2023 IEEE Conference on Standards for Communications and Networking (CSCN)*, 2023, pp. 8–14. [Online]. Available: https://doi.org/10.1109/CSCN60443.2023.10453142

[3] T. Taleb, R. Aguiar, I. Grida Ben Yahia, B. Chatras, G. Christensen, U. Chunduri, A. Clemm, X. Costa, L. Dong, J. Elmirghani, B. Yosuf, X. Foukas, A. Galis, M. Giordani, A. Gurtov, A. Hecker, C.-W. Huang, C. Jacquenet, W. Kellerer, S. Kuklinski, R. Li, W. Liao, K. Makhijani, A. Manzalini, I. Moerman, K. Samdanis, K. Seppänen, D. Trossen, F. Xie, C.-K. Yen, and M. Zorzi, *White paper on 6G networking*, ser. 6G Research Visions. Finland: University of Oulu, Jun. 2020, no. 6. [Online]. Available: https://oulurepo.oulu.fi/handle/10024/36803

[4] M. A. Uusitalo, P. Rugeland, M. R. Boldi, E. C. Strinati, P. Demestichas, M. Ericson, G. P. Fettweis, M. C. Filippou, A. Gati, M.-H. Hamon, M. Hoffmann, M. Latva-Aho, A. Pärssinen, B. Richerzhagen, H. Schotten, T. Svensson, G. Wikström, H. Wymeersch, V. Ziegler, and Y. Zou, "6G Vision, Value, Use Cases and Technologies From European 6G Flagship Project Hexa-X," *IEEE Access*, vol. 9, pp. 160 004–160 020, 2021. [Online]. Available: https://doi.org/10.1109/ACCESS.2021.3130030

[5] R. Asensio-Garriga, A. M. Zarca, and A. Skarmeta, "A Multi-stakeholder Cloud-continuum framework for 6G Networks security & service management," in *2023 IEEE 12th International Conference on Cloud Networking, CloudNet 2023*. Institute of Electrical and Electronics Engineers Inc., 2023, pp. 466–471. [Online]. Available: https://doi.org/10.1109/CloudNet59005.2023.10490020

[6] J. Ellingwood, "An Introduction to Kubernetes," accessed: 2024-09-09. [Online]. Available: https://www.digitalocean.com/community/tutorials/an-introduction-to-kubernetes

[7] "free5GC GitHub repository," access [2024-09-11]. [Online]. Available: https://github.com/free5gc/free5gc

[8] "UERANSIM GitHub repository," access [2024-09-11]. [Online]. Available: https://github.com/aligungr/UERANSIM

[9] I. Leyva-Pupo and C. Cervelló-Pastor, "Efficient solutions to the placement and chaining problem of User Plane Functions in 5G networks," *Journal of Network and Computer Applications*, vol. 197, 2022. [Online]. Available: https://doi.org/10.1016/j.jnca.2021.103269

[10] S. Chen, J. Chen, and H. Li, "Joint optimization of UPF placement and traffic routing for 5G core network user plane," *Computer Communications*, vol. 216, pp. 86–94, 2024. [Online]. Available: https://doi.org/10.1016/j.comcom.2023.12.029

[11] H. T. Nguyen, T. Van Do, and C. Rotter, "Scaling UPF Instances in 5G/6G Core With Deep Reinforcement Learning," *IEEE Access*, vol. 9, pp. 165 892–165 906, 2021. [Online]. Available: https://doi.org/10.1109/ACCESS.2021.3135315

[12] B. G. S. Costa, J. Bachiega, L. R. de Carvalho, and A. P. F. Araujo, "Orchestration in Fog Computing: A Comprehensive Survey," *ACM Computing Surveys (CSUR)*, vol. 55, pp. 1 – 34, 2022. [Online]. Available: https://doi.org/10.1145/3486221

[13] J. Kumar, J. K. Samriya, M. Bolanowski, A. Paszkiewicz, W. Pawłowski, M. Ganzha, K. Wasielewska-Michniewska, B. Solarz-Niesłuchowski, M. Paprzycki, I. L. Úbeda, and C. E. Palau, "Towards 6G-Enabled Edge-Cloud Continuum Computing - Initial Assessment," in *Advanced Communication and Intelligent Systems*, R. N. Shaw, M. Paprzycki, and A. Ghosh, Eds. Cham: Springer Nature Switzerland, 2023, pp. 1–15. [Online]. Available: https://doi.org/10.1007/978-3-031-25088-0_1

[14] N. Nethercote, P. J. Stuckey, R. Becket, S. Brand, G. J. Duck, and G. Tack, "MiniZinc: Towards a Standard CP Modelling Language," in *Principles and Practice of Constraint Programming – CP 2007*, C. Bessière, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 529–543. [Online]. Available: https://dl.acm.org/doi/10.5555/1771668.1771709

[15] Q. Huangfu and J. Hall, "Parallelizing the dual revised simplex method," *Mathematical Programming Computation*, vol. 10, no. 1, pp. 119–142, Mar. 2018. [Online]. Available: https://doi.org/10.48550/arXiv.1503.01889

[16] A. Bęben, M. Sosnowski, W. Jóźwiak, J. Woźniak, K. Gierłowski, M. Hoeft, M. Natkaniec, P. Boryło, B. Belter, M. Furmann, P. Schauer, Łukasz Falas, A. Warzyński, I. Michalski, and D. Więcek, "5G National Laboratory: Perspective and Research Directions," *Przegląd Telekomunikacyjny - Wiadomości Telekomunikacyjne*, vol. 4, pp. 33–42, 2024. [Online]. Available: https://doi.org/10.15199/59.2024.4.4

[17] "Towards5GS-helm GitHub repository," access [2024-09-11]. [Online]. Available: https://github.com/Orange-OpenSource/towards5gs-helm