2024 20th International Conference on Network and Service Management (CNSM)

Budget-Aware Resource Pricing in Cloud and Edge Computing Continuum

Akram Boutouchent
EURECOM
Sophia Antipolis, France
akram.boutouchent@eurecom.fr

Karim Boutiba
EURECOM
Sophia Antipolis, France
karim.boutiba@eurecom.fr

Adlen Ksentini
EURECOM
Sophia Antipolis, France
adlen.ksentini@eurecom.fr

***Abstract*—The emergence of new computing paradigms such as Edge Computing, Fog Computing, and Far-Edge Computing is driven by the increasing demands of modern applications. Together, these paradigms form the Cloud-Edge Computing Continuum (CECC), presenting new challenges in resource allocation and incentive-driven interactions. New stakeholders are joining the business market to make a profit by selling their services (i.e., infrastructure resources, applications, or virtual resources). These actors, namely, infrastructure providers and service providers, have conflicting goals in terms of making a profit. There is a need to study and model the business interaction between these actors, especially considering the distributed nature of continuum. In this paper, we tackle the resource allocation and pricing problem in the context of CECC. We first propose a system model of the incentive interactions between actors of the continuum, where the price of resources varies based on different factors. Then, we formulate a budget-aware resource bidding problem where the objective is to jointly maximize the budget of a service provider and minimize Service Level Agreement (SLA) violations. To address this challenge, we propose a Deep Reinforcement Learning (DRL) approach that efficiently balances budget expenditure and SLA compliance. Our experimental results demonstrate that the proposed method effectively achieves a favorable trade-off between budget management and SLA satisfaction.**

***Index Terms*—Edge Computing, Cloud and Edge Computing Continuum, Resource Pricing, Deep Reinforcement Learning**

I. Introduction

The emergence of latency-sensitive and bandwidth-intensive applications imposed heavy expectations on the communication infrastructure. These new services, such as *Metaverse* [1], *Holographic Communication* [2], and *Autonomous Vehicles* [3], require considerable improvements in the overall networking infrastructure. Consequently, this infrastructure evolved to integrate with caching and computing services to satisfy these requirements. This resulted in the emergence of new paradigms that integrate computing and caching in different parts of the network, intending to address the latency issues and network bottlenecks. *Edge Computing* [4], for instance, is a paradigm that was proposed to bring computing resources closer to the user, which alleviates the challenges of network backbone limitations and high latency of cloud services. Following this strategy, *Cloudlets* [5], *Multi-Access Edge Computing (MEC)* [6], *Fog Computing* [7], and *In-Network Computing (INC)* [8] were proposed as paradigms that follow the same goal, but place computing and caching resources at different places of the network. However, the stringent service requirements of new and innovative applications necessitate the use of multiple computing paradigms to meet these requirements. To this end, a promising approach is envisioning a network-wide architecture, utilizing a continuum of resources across the network, and combining the capabilities of edge servers, the cloud, and end devices to create a hierarchical paradigm known as the *Cloud and Edge Computing Continuum (CECC)* [9].

The CECC brings together diverse actors and stakeholders within a unified infrastructure to meet users' latency and quality of experience requirements. This heterogeneous environment, with varying stakeholder goals and incentive mechanisms, presents significant challenges in business interactions. Given that some infrastructure providers operate within distinct geographical or administrative domains, a service provider may need to engage with multiple providers across different network points to optimize revenue and SLA fulfillment. Modeling these interactions across the entire infrastructure continuum is complex, necessitating intelligent strategies for managing incentives and interactions with various continuum actors.

The network incentive and resource pricing problem has been widely investigated by the research community. Based on the literature, the incentive interaction is generally modeled using Auction Theory and Game Theory. In the scope of CECC, the literature is limited to contributions addressing mainly pricing at the edge computing level, considering cloud, edge servers, and the end users [10], [11]. To our knowledge, there are still limited works that address the pricing problem considering the whole continuum. Additionally, very few works address the challenge by considering the dynamicity of user requests, and service providers. In this paper, we tackle the resource pricing problem considering the whole continuum. We first model the incentive interactions between service providers and infrastructure providers by considering a dynamic environment with varying user requests. We formalize the joint problem of SLA violation minimization and maximization of revenue. Then, we propose a deep reinforcement learning algorithm to solve the problem of jointly maximizing the service provider's budget while minimizing SLA violations. The contributions of our paper can be summarized as follows:

978-3-903176-66-9 ©2024 IFIP

- We model the incentive environment considering the whole cloud and edge computing continuum. We leverage concepts from auction theory to make service providers bids on different resource providers.
- We formulate the resource pricing problem with a joint objective of maximizing the service provider's budget and minimizing SLA violations.
- We propose a Deep Reinforcement Learning approach to address the problem with a well-defined observation, action, and reward space to alleviate high-dimensionality issues that impact the learning stability.
- Finally, we conduct simulation experiments to evaluate the performance of our proposed method.

The remainder of this paper is organized as follows. Section II reviews the related work. In Section III, we present our system model and formulate the optimization problem. We describe our DRL approach in Section IV. Section V analyses the performance of our proposal. Finally, Section VI concludes this work and outlines potential future directions.

II. Related Works

The emergence of edge computing has enabled various applications and improved network performance but also introduced new challenges, particularly in pricing, incentives, and economics within the Cloud-Edge Computing Continuum. While research has addressed these challenges using approaches such as Auction Theory, Game Theory, and Machine Learning, there remains a need for further exploration.

Auction theory [12], a popular economic approach, is leveraged to address the economic challenges in edge computing. Specifically, auction-based mechanisms are promising since they can fairly and efficiently allocate limited resources of sellers to buyers in a trading form at competitive prices. Effective auction mechanisms ensure key properties such as truthfulness, budget balance, and economic efficiency [13]. In an auction theory model applied to edge computing, the sellers are the infrastructure providers, and the buyers (bidders) are service providers or users (end devices). For instance, in the work of [14], authors tackle the issue of maximizing social welfare in MEC systems using a combinatorial auction mechanism called G-ERAP, which allocates virtual machines to end devices. In the same context, Double auctions, suitable for many-to-many scenarios, are explored in [11], where a single-round auction allocates resources based on device preferences and edge server capacities, mediated by a trusted third party. Similarly, the authors in [15] propose a multi-participant double auction model where infrastructure providers sell resources to service providers, facilitated by an auctioneer to ensure fairness. They introduce binary search-based algorithms to optimize resource allocation and ensure bidder truthfulness.

Alternatively, game theory, and especially Stackleberg games, have been heavily leveraged to model the incentive interactions between stakeholders. For instance, in [10], a dual Stackelberg game is employed to design a pricing scheme for IoT applications involving cloud providers, multi-edge infrastructure providers, and end devices. The cloud provider sets the initial price, followed by the edge providers and end users, who adjust their strategies based on these prices. A double-label radius KNN algorithm is used to optimize pricing efficiency. Similarly, [16] models resource allocation in edge computing through a Stackelberg game, where edge providers seek profit maximization, and end devices aim to maximize utility. The problem is solved using an iterative algorithm. In [17], three dynamic pricing mechanisms (BID-proportional, uniform, and fairness-seeking) are analyzed to guide edge service providers on optimal pricing strategies. A sharing economy-based model is proposed in [18], where excess resources are shared among users, leading to improved resource utilization. Additionally, [19] introduces a market-based framework to allocate resources from geographically distributed edge nodes to competing services, achieving market equilibrium through strategic pricing.

In another context, machine learning, especially Reinforcement Learning (RL) proved to be very efficient in tackling problems in complex environments. While each RL agent can represent a stakeholder in the game, a multi-agent scenario might be suitable to maximize the fairness and utility of each actor. This has been experimented with by the authors of [20]. The work considers a 5G network scenario where virtual network functions are allocated across multiple domains. The authors propose an auction-based approach for inter-domain resource allocation using a distributed multi-agent RL solution.

Following the literature review conducted in the preceding section, a noticeable gap is identified in the contextual focus on CECC. Many of the works that address the challenge of resource pricing consider cloud computing and edge computing as separate entities, whereas we consider the whole continuum. Additionally, most of the works primarily address challenges from the perspective of infrastructure providers. In our work, we consider a service provider that needs to allocate resources to satisfy its users and generate revenue. Finally, some works consider users as participants in the auction, which is not realistic, since the users interact only with a service provider. Our approach involves the interaction of various stakeholders, including cloud providers, service providers, and fog/far-edge infrastructure providers. In the following section, we propose our system model and define a resource pricing problem that aims to jointly maximize revenue and minimize SLA violations.

III. System Model And Problem Formulation

A. System Model

In this section, we describe the system model that we consider in such a complex environment. Fig. 1 gives a high-level representation of the incentive interactions between the different actors. The system operates as an auction, where infrastructure providers act as sellers, offering resources at a base price, while service providers (bidders) submit bids for specific infrastructure. The auction is managed by the Cloud and Edge Computing Continuum Manager (CECCM), a central entity responsible for coordinating interactions, resource management, and federation across the continuum. CECCM abstracts

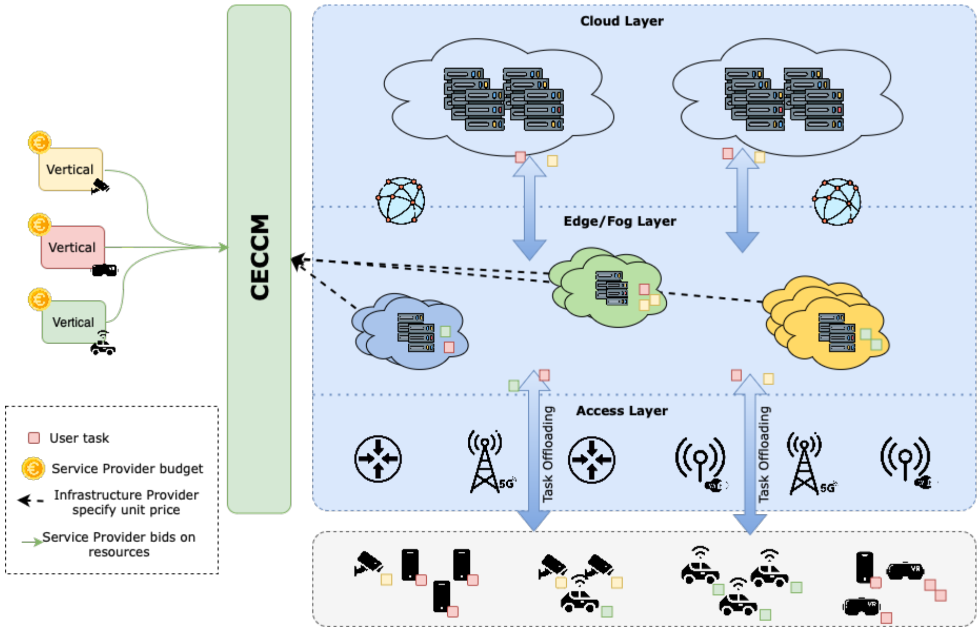


Fig. 1: Cloud and Edge Computing Interactions.

management processes into a unified layer, ensuring standardized interactions via APIs from organizations like GSMA OPG and CAMARA. Inspired by initiatives like GAIA-X [21], which promotes secure and trusted data exchange, the CECCM plays a neutral role as auctioneer, handling payments and resource allocation. It determines the auction winner for each infrastructure provider and allocates resources accordingly. Once services are deployed, users offload their requests to access these services. Service providers aim to efficiently allocate resources by bidding within their budget constraints. The following sections describe the detailed system model, focusing on infrastructure and service providers.

1) Infrastructure: We consider a Cloud and Edge Computing Continuum infrastructure composed of heterogeneous infrastructure providers. Each entity encompasses a set of computing resources centralized in data centers located in a specific region. We consider three types of infrastructure providers, namely, Cloud providers, Edge/Far-Edge providers, and Fog providers. Consequently, we model the infrastructure as a graph $G = (R, L)$ such that R represents the set of infrastructure providers (nodes of the graph), and L represents the network links. Each node r in the infrastructure has a set of resources $N_r = \{N_{r,v} \mid v \in V\}$ such that $N_{r,v}$ represents the amount of resources of type v of node r. For instance, if it is a CPU resource then the value represents the cycles per time. A link $l = (p, r)$ represents a physical or virtual link between the nodes p and r. It is identified by a bandwidth amount E_l^{Bw} and a propagation delay E_l^{Pd}. Fig. 2 gives a high-level representation of the infrastructure nodes, links as well as the auction system. It is worth mentioning that the infrastructure includes a set of special nodes referred to as access nodes (base stations, wireless access points) that don't have resources ($N_r = \{0 \mid v \in V\}$). At the start of the auction, the infrastructure providers announce their base prices $P_{r,v}$ which represents the price per unit for a certain type of resource. Table I presents the main notations used in this work.

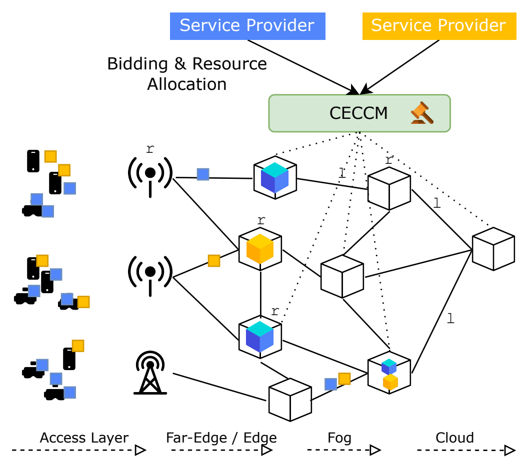


Fig. 2: High-Level Representation of the auction system.

2) Service Provider: Let's consider a set of service providers denoted as $\mathcal{S} = \{s_1, s_2, ...s_n\}$, where $|\mathcal{S}| = S$. We assume that each service provider has a set of users, namely subscribers. The service provider needs therefore to allocate a set of resources to satisfy their SLA requirements. For each round, the infrastructure providers announce their unit prices, this value increases whenever we get closer to the access nodes. Let us define a set of users as $\mathcal{U}$, assuming that each user can utilize only one service at any given time. The users of a specific service provided by the service provider s are represented as $\mathcal{U}_s$, where $\mathcal{U} = \{\mathcal{U}_s \mid \forall s \in \mathcal{S}\}$.

Symbol	Description
$\mathcal{S}$	Set of service providers $\{s_1, s_2, ...s_n\}$
R, L, V	Set of nodes, links and resource types
$N_{r,v}, N'_{r,v}$	Total amount of resources of type v in node r and the available resources
E_l^{Bw}, E_l^{Pd}	Bandwidth and propagation delay of link l
$P_{r,v}$	Base price for a unit of v resource on the infrastructure provider r
$\mathcal{U}_s$ Q_u^s	Set of users of the service provider s Request of user u to service s_i
M^{DATA}, M^{CPU}	Average data size of a request message, and average work, measured in CPU instructions, to process a request message.
λ_s	Service Level Agreement's deadline of service provider s
$\mathcal{B}_s$, $b_s(.)$	Initial budget and remaining budget of a service provider s
$F_{s,v}(.)$, $l_{s,r}$	Function to calculate amount of resource required based on the request rate, and the user load of service provider s on node r
$\mathbf{X}$, $x_{r,v}$	The bidding strategy, and the bid value on node r and resource type v
$W_r^s(\mathbf{X})$	Result of auction of service provider s on node r based on its bidding strategy
$a_{r,v}^s(\mathbf{X})$	Allocated budget of service provider s on node r based on the bidding strategy
$\theta_s(\mathbf{X})$, $D_s(r, \mathbf{X})$	The average deadline violation of all deployments, and the average deadline violation on a node r

TABLE I: System Model Notations

The service provider's user sends requests to the deployed services. These requests are characterized by the number of resources required to process the request and the data size of the request message. We define the user request as follows $Q_u^s = (M^{DATA}, M^{CPU})$, where M^{DATA} is the average data size of a request message, and M^{CPU} is the average work, measured in CPU instructions, to process a request message. The users of a certain service provider have agreed on a certain latency that we define as λ_s. The users' requests need to respect this agreement and the service provider's goal is to minimize the SLA violations. On the other hand, Each service provider enters the market with an initial budget $\mathcal{B}_s$, aiming to maximize this budget and maintain market presence. The service provider is also subject to SLA violation constraints, therefore the SLA violations need to be minimized. The service providers engage in a bidding process with different infrastructure providers to allocate resources and provision services to serve users' requests, based on the prices proposed by the infrastructure providers. Similarly, infrastructure providers seek to maximize their budgets by offering resources at competitive prices. We consider the time divided into a set of periods, such that the price to pay is defined per period. This means that the budget represents the amount of money that the service provider will be paying each period. This is very close to cloud computing subscriptions, where resource prices are paid on a per-hour price.

B. Problem Formulation

At the start of a certain period, the infrastructure providers announce their base price for the resources they are selling as well as the amount of available resources to the CECCM. The service providers need, therefore, to compete for these resources by participating in the auction to allocate resources for a given price. The goal of service providers is to satisfy their users while minimizing their budget spending. The service providers need to decide upon the unit price of resources to bid and the amount of resources to allocate. To simplify the process of resource allocation, we model the resource demand as a function of the request rate [22]. $Fs = \{F_{s,v}(x) \mid v \in V\}$ describes the scalable resource demands. More specifically, $F_{s,v}(x)$ is a function expressing the amount of resource $v \in V$ that must be allocated to an instance of the service s to handle an incoming load x, which is the average request arrival rate at this instance. Once a request is sent by the user, it gets routed to the closest computing node based on the network distance with the user. The service provider's decision will mainly concern the unit price to bid on each node of the infrastructure. The resource bidding scheme is referred to as $\mathbf{X} = \{x_{r,v} \mid v \in V, r \in R\}$ such that $x_{r,v}$ represents a unit price that service provider is willing to pay for an infrastructure provider r over the usage of resources.

The service provider's bidding strategy will affect the amount of money that it will spend. The auction will decide on the winner based on the highest bid price. The service provider has an estimated budget to be spent, which is the maximum budget that it can spend $\mathcal{B}_s$. The budget that is left is calculated based on the bids it won. $W_r^s(\mathbf{X})$ represents whether a service provider has won the auction on a certain infrastructure provider r. The service provider will be allocated resources from the bids it wons depending on the budget. The CECCM will analyze the capacity of a service provider to pay and cancel its bid in case the budget is not available. Therefore $W_s(\mathbf{X})$ represents the set of auction results of a service provider based on the bidding strategy such that $W_s(\mathbf{X}) = \{W_r^s(\mathbf{X}) \mid r \in R \ , \ W_r^s(\mathbf{X}) \in \{0, 1\}\}$. On the other hand, once a service provider wins a bid on a certain infrastructure provider, it will allocate most of the resources that satisfy its resource demand. We denote the amount of budget allocated by service provider s on node r as $a_{r,v}^s(\mathbf{X})$. The allocated budget is calculated based on the load received at a node r denoted as $l_{s,r}$. We also denote the remaining budget for the service provider after the auction as $b_s(\mathbf{X})$. Equation 1a specifies how the final budget will be calculated based on the initial budget and the bidding strategy.

$$b_s(\mathbf{X}) = \mathcal{B}_s - \sum_{r \in R} \sum_{v \in V} W_r^s(\mathbf{X}) \ a_{r,v}^s(\mathbf{X}) \tag{1a}$$

$$a_{r,v}^s(\mathbf{X}) = \min \{N'_{r,v}, F_{s,v}(l_{s,r})\} \ x_{r,v} \tag{1b}$$

On the other hand, the service provider is constrained by deadline violations. The deadline violation $\theta_s(\mathbf{X})$ is calculated based on the average deadline violation of all requests to each deployed instance. Equation 2a calculates the average deadline violation over all the deployed instances of service s. $D_s(r, \mathbf{X})$ represents the weighted average deadline violation of the instance deployed on node r. Moreover, $\mathbb{1}(x)$ is an indicator function equal to 1 if $x > 0$ and 0 otherwise.

$$\theta_s(\mathbf{X}) = \frac{1}{|\mathcal{R}|} \sum_{r \in \mathcal{R}} D_s(r, \mathbf{X})\ W_s^r(\mathbf{X}) \tag{2a}$$

$$D_s(r, \mathbf{X}) = \frac{\mathbb{1}\left(RT(s,r) - \lambda_s\right) l_{s,r}}{\sum_{r \in \mathcal{R}} l_{s,r} W_s^r(\mathbf{X})} \tag{2b}$$

Where $RT(s,r)$ is the average response time of an instance of service s deployed in node r.

We can now formulate the joint resource pricing problem for a service provider that aims to minimize the budget spending and minimize QoS violations as follows:

$$\mathcal{P} : \min_{\mathbf{X}} \quad w\theta_s(\mathbf{X}) + (1-w)(\mathcal{B}_s - b_s(\mathbf{X})) \tag{3a}$$

$$\text{s.t.} \quad \sum_{v \in V} x_{r,s} N'_{r,v} \ < \mathcal{B}_s \qquad \forall r \in \mathcal{R} \tag{3b}$$

$$x_{r,s} \geq 0 \qquad \forall r \in \mathcal{R} \tag{3c}$$

Where $w \in [0,1]$ is a constant weight that defines the importance of both objectives. Constraint (3b) ensures that the service provider can not make bids that go above its budget. Constraint (3c) restricts bidding negative values for each node of the infrastructure.

IV. A Deep Reinforcement Learning Approach

The resource pricing problem can be framed as an Integer Linear Problem (ILP), which is often difficult to solve with traditional methods due to complexities like dynamic end-user behavior (e.g., mobility) and variable delays based on deployment. To overcome these challenges, we propose a deep reinforcement learning (DRL) approach to solve problem $\mathcal{P}$, avoiding the need for explicit modeling of these parameters.

Reinforcement learning excels in dynamic environments, adapting to unseen scenarios. By leveraging this, a service provider can delegate financial decisions to the DRL agent, which aims to maximize revenue while minimizing deadline violations. Service providers will compete for resources, seeking an optimal balance between budget expenditure and SLA compliance.

The action space that the DRL agent will explore includes $\mathbf{X}$, the set of bidding prices for each infrastructure provider. This means that the service provider needs to take into consideration the whole continuum nodes to make a decision. Such an approach can be improved, leading to reduced action space for the DRL agent and resulting in a stable learning. To reduce the action space of the DRL agent, we follow a progressive auction system. This can alleviate issues due to large action space that can affect the model's learning. At the start of each time step t, we explore the infrastructure node by node. The service providers need to iteratively make a bid on the resources of an infrastructure provider. The episode will end when we arrive at the last node. Consequently, the CECCM will decide on the winner of the auction and allocate resources. In what follows, we present our state, action, and reward space.

A. State Space

The agent will explore the topology node by node, the current node is denoted as r_t. Consequently, the state space S of the agent includes the following:

- $N'_{r_t} = \{N'_{r_t,v} |\ v \in V\}$, which represents the amount of available resources on the current node.
- $DIS_{r_t} = \{DIS(r_t, r) |\ r \in R_{access}\}$, which represents the network delay between the current node and all the access nodes that users connect through.
- $F_t^s = \{F_{t,v}^s |\ v \in V\}$, which represents the current resource demand.
- λ_s, the service provider's deadline, specified in the SLA.
- θ_t^s represents the current deadline violation with the current won bids and allocated resources. This value is calculated based on the users' requests and current allocated services. Every time the service provider allocates resources the deadline violation is prone to change. It either decreases or stays the same. It decreases only if the service was allocated in a better node that can satisfy users more than the other previously deployed services.
- b_t^s, the current remaining budget, based on the last bids won and allocated resources.
- $a_{r_{t-1},v}^s$, representing the latest payment made. This is calculated based on the auction result of the last node and the bid price.
- $TYPE_{r_t}$, this is the node type that can be either edge/far-edge node, fog node or cloud node.
- $P_{r_t,v}$, which is the base price announced by the current infrastructure provider.

B. Action Space

The agent's action is to select a bidding price for each node progressively at each time step. The bid price needs to be, therefore, greater than 0 and less than its remaining budget divided by the amount of resources it will allocate. For simplicity, we define a $[0, a]$ space for the service provider just to alleviate the problem of dynamic action space. The variable a denotes the upper limit of this action space, corresponding to the maximum price the agent is allowed to bid.

C. Reward Space

To help the agent learn a proper policy, we design a progressive reward function. The reward function we are employing is presented as follows:

$$\mathcal{R}(t) = wA(t) + (1-w)B(t) \tag{4a}$$

$$A(t) = \begin{cases} (\theta_{t-1}^s - \theta_t^s) & \text{If } W_{r_t}^s = 1 \\ 0 & \text{Otherwise} \end{cases} \tag{4b}$$

$$B(t) = \begin{cases} \frac{1}{1+(\mathcal{B}_s - b_t^s)} & \text{If } t = |T| - 1 \\ 0 & \text{Otherwise} \end{cases} \tag{4c}$$

The instant reward that the agent receives after taking an action is $\mathcal{R}(t)$. It is composed of two main building blocks. First part $A(t)$ represents the difference between the previous deadline violation θ^s_{t-1} and the new deadline violation θ^s_t. The value means that the action realized by the agent in the last step which mean deploying a new service resulted in an improved SLA satisfaction. If the service provider didn't won the bid, then this part of the reward will be 0. We leveraged this in order to encourage the agent to bid higher and satisfy the user SLAs. On the other hand, $B(t)$ which is the second part of the reward include $(\mathcal{B}_s - b_s(t))$ which refers to the amount of spent budget at time step t. The service provider needs to minimize this value. Both $A(t)$ and $B(t)$ have values in the range $[0, 1]$, the $B(t)$ is given to the agent at the end of the episode, contrary to $A(t)$ which is given at each time step. We leverage w as a weight variable in order to balance between budget minimization and SLA satisfaction.

TABLE II: Experimentation Parameters

Parameter	**Value**
CPU Resource $N_{r,v}$ (×$10^9 IPS$)	Edge/Far-Edge: [5,10], Fog: [10,50], Cloud: 1000
Bandwidth E_l^{Bw} (Mbps)	Edge, Far-Edge, Fog: [100, 200], Cloud: [1000]
Propagation delay E_l^{Pd} (ms)	Edge, Far-Edge, Fog: [1, 10], Cloud: [10,15]
Base Price $P_{r,v}$ (10^-2 $\$/h$)	Edge/Far-Edge: [40,70], Fog: [20,40], Cloud: [10]
Deadline λ_s (s)	[0.04, 0.08]
Initial service provider's budget $\mathcal{B}_s$ ($/h)	[3, 4.5]
User request rate	1.0 request/s
User distribution	Zipf$(k), k = [1.4, 3.0]$
Data M^{DATA} (Kb)	$[1, 10]$
CPU Work M^{CPU}	$[1, 10] \times 10^6$ CPU Instructions
DRL Hyperparameter	
Learning rate	0.0004
Discount factor	$\gamma = 0.99$
Batch size	32
Total timesteps	3×10^4

$[a, b]$: a value is chosen randomly within this range.

V. Performance Evaluation

A. Experiment Setup

To train and evaluate the performance of our proposed system, we use YAFS [23], a fog computing simulator that models infrastructure as a graph and simulates network routing, service computation, and latency. Experiments and training were conducted on a PC with a Core-i7 CPU and 32GB of RAM. For real-world topology simulation, we utilized Zoo topologies [24], which provides network data from publicly available sources. The deep reinforcement learning agent was built using the Stable Baselines library [25]. The training process begins by generating a topology from Zoo topologies, which is augmented with base prices and compute resources for each node. User requests, modeled with a Zipf distribution, simulate varying service demands. Service providers compete for infrastructure resources, with requests routed to the nearest compute node that hosts the service. In our setup, 10 service providers bid on resources—9 bid at the base price with random variations, while 1 follows a specific policy. We set the value of w to 0.8, prioritizing deadline violations, which led to improved performance under higher user loads. Experiment parameters are summarized in Table II, and we evaluate the following policies:

- **Cloud-only** The service provider bids a higher price on the cloud nodes.
- **Edge-only** The service provider follows a greedy approach, by bidding on the edge nodes primarly.
- **Random** The service provider follows a random policy and bids randomly with a random price.
- **DRL-DQN** The deep reinforcement learning agent based on Deep Q-Network.

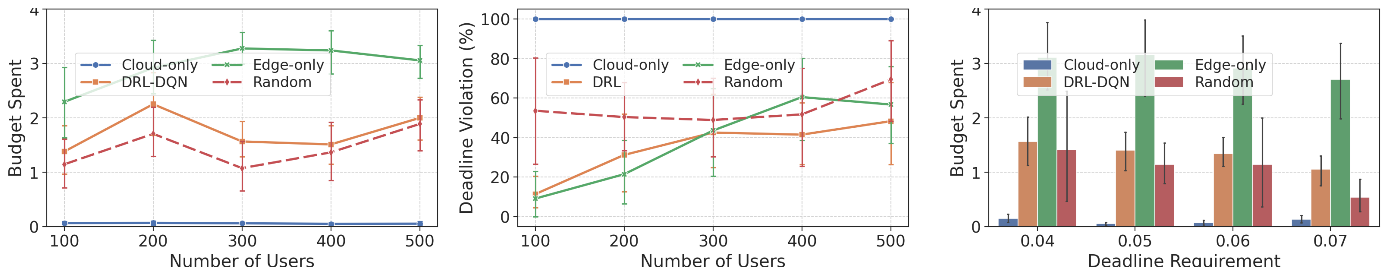
B. Results and Discussion

The experimentation was conducted on a CECC infrastructure with 20 nodes of different types, varying both the number of users and their deadline requirements. Below, we discuss the results and performance of our approach.

Fig. 3a shows the budget spent as the number of users increases. For each user count, we ran 10 trials with varying deadlines and averaged the results. The cloud-only agent performs best in terms of budget minimization, as it bids only on inexpensive cloud nodes. In contrast, the edge-only agent is the most costly, frequently exhausting its budget since edge resources are more expensive. The DRL agent strikes a balance, dynamically bidding based on user requests and resource types, minimizing budget use efficiently.

In terms of deadline violation, Fig. 3b illustrates the normalized deadline violation of the users' requests by varying the number of users. The cloud-only agent consistently underperforms, with nearly 100% violations due to high latency from distant cloud nodes. The edge-only agent performs well initially but struggles with high user loads, as limited edge resources force requests into queues. After 400 users, the DRL agent outperforms, balancing resource allocation across edge and fog nodes, thereby maintaining a better trade-off between budget and performance.

Finally, in Fig. 3c, we illustrate the amount of spent budget based on the deadline requirement specified in the experimentations. These results aggregate all the experimentations with different numbers of users. As deadlines are relaxed, the DRL agent spends less, reallocating resources across more cost-effective nodes, unlike the edge-only agent, whose costs remain fixed. Overall, the DRL agent optimizes budget use while reducing deadline violations, effectively learning a policy that balances both factors based on user demand and infrastructure pricing.



(a) Budget spent by service provider. (b) Normalized deadline violation. (c) Budget spent with varying deadline requirements.

Fig. 3: Experiment results considering 20 nodes and a varying number of users

VI. Conclusion and Future Works

In this paper, we tackled the resource pricing problem within the CECC framework. We modeled an auction system where service providers bid on infrastructure resources to meet user demands. We formalized the joint problem of budget optimization and deadline violation minimization, proposing a deep reinforcement learning algorithm that demonstrated effective results. Future work will explore multi-agent reinforcement learning to simulate a competitive market, as well as hierarchical reinforcement learning, where smaller agents manage decisions across different infrastructure regions.

Acknowledgment

This work was partially supported by the European Union's Horizon Research and Innovation program under AC^3 project and grant agreement No 101093129.

References

[1] Y. Wang, Z. Su, N. Zhang, R. Xing, D. Liu, T. H. Luan, and X. Shen, "A survey on metaverse: Fundamentals, security, and privacy," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 1, pp. 319–352, 2023.

[2] A. Clemm, M. T. Vega, H. K. Ravuri, T. Wauters, and F. D. Turck, "Toward truly immersive holographic-type communication: Challenges and solutions," *IEEE Communications Magazine*, vol. 58, no. 1, pp. 93–99, 2020.

[3] R. Hussain and S. Zeadally, "Autonomous cars: Research results, issues, and future challenges," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1275–1313, 2019.

[4] W. Z. Khan, E. Ahmed, S. Hakak, I. Yaqoob, and A. Ahmed, "Edge computing: A survey," *Future Generation Computer Systems*, vol. 97, pp. 219–235, Aug. 2019.

[5] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for vm-based cloudlets in mobile computing," *IEEE Pervasive Computing*, vol. 8, no. 4, pp. 14–23, 2009.

[6] D. S. et al, ""Mobile edge computing a key technology towards 5G"," Tech. rep. White Paper. Sophia Antipolis, France, ETSI, 2015, p. 16.

[7] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, ser. MCC '12. New York, NY, USA: Association for Computing Machinery, 2012, p. 13–16.

[8] S. Kianpisheh and T. Taleb, "A survey on in-network computing: Programmable data plane and technology specific applications," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 1, pp. 701–761, 2023.

[9] A. Maia, A. Boutouchent, Y. Kardjadja, M. Gherari, E. G. Soyak, M. Saqib, K. Boussekar, I. Cilbir, S. Habibi, S. O. Ali, W. Ajib, H. Elbiaze, O. Erçetin, Y. Ghamri-Doudane, and R. Glitho, "A survey on integrated computing, caching, and communication in the cloud-to-edge continuum," *Computer Communications*, vol. 219, pp. 128–152, 2024.

[10] T. Wang, Y. Lu, J. Wang, H.-N. Dai, X. Zheng, and W. Jia, "Eihdp: Edge-intelligent hierarchical dynamic pricing based on cloud-edge-client collaboration for iot systems," *IEEE Transactions on Computers*, vol. 70, no. 8, pp. 1285–1298, 2021.

[11] Y. Yue, W. Sun, and J. Liu, "A double auction-based approach for multi-user resource allocation in mobile edge computing," in *2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*, 2018, pp. 805–810.

[12] V. Krishna, *Auction Theory*. Academic Press, 2002.

[13] H. Qiu, K. Zhu, N. C. Luong, C. Yi, D. Niyato, and D. I. Kim, "Applications of auction and mechanism design in edge computing: A survey," *IEEE Transactions on Cognitive Communications and Networking*, vol. 8, no. 2, pp. 1034–1058, 2022.

[14] T. Bahreini, H. Badri, and D. Grosu, "An envy-free auction mechanism for resource allocation in edge computing systems," in *2018 IEEE/ACM Symposium on Edge Computing (SEC)*, 2018, pp. 313–322.

[15] J. Huang, S. Li, L. Yang, J. Si, X. Ma, and S. Wang, "Multi-Participant Double Auction for Resource Allocation and Pricing in Edge Computing," *IEEE Internet Things J.*, pp. 1–1, 2023.

[16] Y. Chen, Z. Li, B. Yang, K. Nai, and K. Li, "A stackelberg game approach to multiple resources allocation and pricing in mobile edge computing," *Future Generation Computer Systems*, vol. 108, pp. 273–287, 2020.

[17] B. Baek, J. Lee, Y. Peng, and S. Park, "Three dynamic pricing schemes for resource allocation of edge computing for iot environment," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4292–4303, 2020.

[18] M. Siew, D. Cai, L. Li, and T. Q. S. Quek, "Dynamic pricing for resource-quota sharing in multi-access edge computing," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 2901–2912, 2020.

[19] D. T. Nguyen, L. B. Le, and V. Bhargava, "Price-based resource allocation for edge computing: A market equilibrium approach," *IEEE Transactions on Cloud Computing*, vol. 9, no. 1, pp. 302–317, 2021.

[20] M. Dieye, W. Jaafar, H. Elbiaze, and R. H. Glitho, "Market driven multidomain network service orchestration in 5g networks," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 7, pp. 1417–1431, 2020.

[21] A. Braud, G. Fromentoux, B. Radier, and O. Le Grand, "The road to european digital sovereignty with gaia-x and idsa," *IEEE Network*, vol. 35, no. 2, pp. 4–5, 2021.

[22] A. M. Maia and Y. Ghamri-Doudane, "A deep reinforcement learning approach for the placement of scalable microservices in the edge-to-cloud continuum," in *GLOBECOM 2023 - 2023 IEEE Global Communications Conference*, 2023, pp. 479–485.

[23] I. Lera, C. Guerrero, and C. Juiz, "Yafs: A simulator for iot scenarios in fog computing," *IEEE Access*, vol. 7, pp. 91 745–91 758, 2019.

[24] S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, and M. Roughan, "The internet topology zoo," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 9, pp. 1765–1775, 2011.

[25] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021.