

# Joint Resource Allocation and Intrusion Prevention System Deployment for Edge Computing

{Chun-Yen Lee, Chia-Hung Lin}, Zhan-Lun Chang, Chih-Yu Wang, Hung-Yu Wei

**Abstract**—Distributed Denial-of-Service (DDoS) attack is critical to latency-critical systems such as Multi-Access Edge Computing (MEC) as it significantly increases the response delay of the victim service. An intrusion prevention system (IPS) is a promising solution to defend against such attacks. Still, there will be a trade-off between IPS deployment and application resource reservation as IPS deployment will reduce the computational resources for MEC applications. In this work, we propose a game-theoretic framework to study the joint computational resource allocation and IPS deployment in the MEC architecture. Given the expected attack strength and end-user demands, we study the pricing strategy of the MEC platform operator (MPO) and purchase strategy of the application service providers (ASPs). The best responses of both MPO and ASPs are derived theoretically. Based on the best responses, we propose an efficient algorithm to derive the Stackelberg equilibrium. The properties and optimality in the efficiency of the equilibrium are analyzed through simulations. The results confirm that the proposed solutions significantly increase the social welfare of the system.



## 1 INTRODUCTION

Multi-access edge computing (MEC) is a promising technique to provide low-latency computing services. MEC platform providers (MPOs) deploy MEC servers at the network edge to offer computational resources for latency-critical tasks. Application service providers (ASPs) purchase those resources to deploy their applications for the end-users (EUs). This architecture serves as the foundation of 5G/B5G killer applications such as Vehicle Automation, AR/VR, etc.

Nevertheless, network attacks have been considered a serious threat to most network services [1]. One of the most common attacks is the Distributed Denial-of-Service (DDoS) attack, which can be categorized into two types: network-level and application-level attack [2]. Network attacks are the most frequent DDoS attacks as they may damage multiple services and applications at the same time. Nevertheless, it requires a significant amount of attackers or bots. On the other hand, the application attacks, which mainly target the application layer, can result in devastating damage to specific ASPs [3] with fewer costs. In the MEC system, the application-level DDoS attacks are performed by many infected devices to send fake requests to the victim service to exhaust the server's resources [4]. Normal end-user requests will be denied afterward because the victim service lacks resources. We observe that this kind of DDoS attack would be critical in the MEC architecture, as the available resources at the MEC servers are usually limited and cannot be expanded, while the deployed applications are mostly latency-critical and cannot tolerate service de-

nial. A defensive measure is crucial to guarantee the service quality under the foreseeable DDoS attacks [5] [6].

Intrusion Prevention System (IPS) is an effective measure to identify and respond to the malicious behaviors performed by attackers [7]. For instance, in DDoS attacks, an IPS instance can recognize and isolate malicious requests from the service requests of normal EUs [8]. Nevertheless, due to the deployment limitation of the network edge environment, the IPS deployed to MEC is likely to be software-based, which means there will be a trade-off between IPS deployment and application resource reservation: the deployment of IPS will reduce the amount of computational resource for applications as the IPS instances occupy some, and thus the service latency may increase accordingly. On the other hand, lacking sufficient IPS will make the application vulnerable to DDoS attacks. Such a trade-off should be considered by the ASP when deploying its application to the MEC system.

We propose a game-theoretic framework to study the joint computational resource allocation and IPS deployment in the MEC architecture from the economic perspective. Specifically, the MPO provides the MEC computational resource in the form of virtual machines (VMs) with a price [9]. The ASPs buy VMs from the MPO to deploy their applications, and the IPS can be optionally deployed if DDoS attacks are expected. In the proposed framework, we form the problem as a Stackelberg game to study 1) the pricing strategy of the MPO and 2) the number of VMs for application and IPS deployment of the ASPs. The ASPs determine the number of IPS to mitigate the DDoS attacks and decide the number of VMs to buy from the MPO. The MPO chooses the price of the VM, which maximizes its profit. The best responses of both MPO and ASPs are derived theoretically, and the corresponding Stackelberg equilibrium can be derived through the proposed algorithm. The simulation results confirm that the proposed solutions

*The authors in curly brackets contributed equally to this work.*

*This work was supported by the National Science and Technology Council under Grant 111-2628-E-001-002-MY3, 112-2218-E-194-005-, and 112-2628-E-002-025-, and the Academia Sinica under Thematic Research Grant AS-TP-110-M07-3.*

significantly increase the social welfare of the system. The main contributions of this work are:

- 1) To the best of our knowledge, we are the first group to examine the tradeoff in IPS deployment for defending against DDoS attacks in the MEC environment. Our proposed game-theoretic approach allows both MPO and ASPs to optimize their configurations for maximum economic benefit.
- 2) We propose a flexible strategy for IPS deployment to protect against DDoS attacks in the MEC system. This involves allowing ASPs to decide on the resource allocation for IPS and the main application service based on the expected attack strength.
- 3) The optimal strategies of both MPO and ASPs are analyzed using the proposed Stackelberg game model, which identifies the equilibrium in polynomial time. The simulation results verify the effectiveness of the proposed solution in maintaining social welfare.

This work is an extended version of our previous work [10]. In this paper, we provide rigorous proof of the convexity of the players' optimization problems in the Game Formulation Section. We further propose a more precise model to capture the characteristics of the IPS VM. We also investigate more complicated cases in the ASP optimization section and provide complete proofs for the best responses of the ASPs. In the MPO optimization section, we proposed a comprehensive mathematical proof of the convexity of the problem and also proved the existence of Stackelberg equilibrium.

## 2 RELATED WORKS

DDoS attacks are considered one of the major threats to network services to be dealt with. Ahmed *et al.* presented a classification framework to detect DDoS attacks using application fingerprints, which are generated by packets' traffic patterns [11]. Santoyo-Gonzalez *et al.* presented a lightweight, platform-independent anomaly detection mechanism based on the BPFabric architecture [12]. Ali *et al.* designed a three-tier intrusion detection and prevention system in the SDN-IoT environment to assure security [13]. A multi-level DDoS mitigation framework [14] was proposed by Yan *et al.* to mitigate DDoS attacks on IoT. Shen *et al.* implemented a two-phase DDoS detection system and showed that it could detect attacks with little overhead [15].

Several studies focus on enhancing intrusion detection to fend off DDoS attacks. Some articles adopted entropy-based methods to identify the DDoS attack and achieve a decent detection rate [16] [17]. Gao *et al.* proposed an onmi intrusion detection system to detect DoS attacks in supervisory control and data acquisition (SCADA) networks [18]. Liu *et al.* developed a Deep Convolution Neural Network Q-learning model to prevent Mirai botnet [19]. Zhang *et al.* built dynamic deployment of classifiers based on a reinforcement learning model to detect DDoS attacks [20].

Several literatures have studied strategies for alleviating DDoS attacks on MEC systems. Tan *et al.* used a novel cooperative defense orchestration mechanism and introduced algorithms to effectively organize the cooperation of defense resources in MEC for DDoS mitigation [21]. Deng *et al.* introduced a Graph Neural Network (GNN)-based Collabora-

tive Deep Reinforcement Learning (GCDRL) model, which mitigates DDoS issues by generating resource allocation strategies on MEC-enabled software-defined networking-based VANET systems [22]. FlowGuard, an intelligent edge defense mechanism against IoT DDoS attacks, utilizing long short-term memory (LSTM) and convolutional neural network (CNN), was proposed by Jia *et al.* [23]. Chang *et al.* proposed an algorithm to allocate resources for Intrusion Detection and Protection Services (IDPS) to mitigate DDoS attacks and enhance Quality-of-Experience (QoE) of applications [24].

Game theory has been applied in wireless systems, such as resource management [25], power control [26], traffic relaying [27], etc. Game theory can also be applied when dealing with intrusion detection systems (IDS). Wang *et al.* proposed an attack-defense game using repeated game approaches on Cyber-Physical Embedded Systems (CPESs) to reduce energy consumption and enhance the detection rate using IDS [28]. Studies aimed at solving the resource allocation problem in wireless systems using game theory. Ma *et al.* proposed to jointly resolve the resource allocation problem of devices, edge nodes, and service providers [29]. The Stackelberg game is widely adopted when solving resource allocation problems. Wang *et al.* formulated a task offloading problem in non-orthogonal multiple access (NOMA)-based MEC using the Stackelberg game [30]. Liu *et al.* applied the Stackelberg game to the resource allocation problem in D2D-enabled vehicular communication systems and maximized the effectiveness of the system [31].

Several works focus on seeking defensive strategies against DDoS in MEC systems through game-theoretic approaches. He *et al.* designed a mechanism to mitigate the impact of DDoS attacks by transferring requests from busy edge servers to their neighbor servers and adopted a game theory model to solve the optimization problem [32]. Liu *et al.* considered the incomplete information about the resource allocation strategy and formulated a Bayesian Q-learning game to resolve the DDoS problems in sensor edge cloud (SEC) [33]. Li *et al.* presented a framework CODE4MEC to coordinate resources and proposed a auction-based schedule algorithm to defend application-layer DDoS attack [34]. Yet, neither of these works considered using IPS to defend against DDoS attacks. In addition, Guo *et al.* proposed a price-based resource orchestration algorithm (PROA) to ensure resource utilization under DDoS attacks [35]. In this work, the system model allows the MEC nodes under attack to request IPS resources from other nodes, and the relationship can be described as followers and leaders in the Stackelberg game. The resource providers (leaders) decide the resources' price first, and the requesters (followers) determine the number of requested resources based on the leaders' actions. Although Guo *et al.* use IPS as the primary solution to defend against DDoS attacks, they did not consider the trade-off in IPS deployments, as the IPS VMs may compete with the resources of the main application service. Additionally, the proper configurations of IPS under different workloads, attack intensity, and IPS efficiency have yet to be studied.

Distinguished from those previous works, we focus on joint IPS deployment and computational resource allocation problems when defending against DDoS attacks. In this

Paper	System	Game-theoretical Approaches	Main DDoS Defense Method	Approaches	Tradeoff Analysis (Service and Defense)
Proposed	Edge	Stackelberg	IPS	Resource allocation	Yes
Yan <i>et al.</i> [14]	SDN-IoT	Not game related	Multi-level SDN framework managing IoT	Propose an SDN framework	No
Tan <i>et al.</i> [21]	Edge	Not game related	IPS	Resource allocation	No
Deng <i>et al.</i> [22]	SDVN	Not game related	GNN-Based resource provisioning model	Resource allocation	No
Jia <i>et al.</i> [23]	Edge	Not game related	Machine learning based detection	Improve detection rate	No
Chang <i>et al.</i> [24]	Edge	Not game related	IDPS	Resource allocation	Yes
Wang <i>et al.</i> [28]	CPES	Repeated game	IDS	Analyze strategies from attacks and defenses	No
He <i>et al.</i> [32]	Edge	Potential Game	Offload tasks to adjacent edge servers	Resource allocation	No
Liu <i>et al.</i> [33]	SEC	Bayesian Q-learning game	Cooperation between sensors and edgeVM	Resource allocation	No
Li <i>et al.</i> [34]	Edge	Auction-based scheduling	IPS	Proposed a cooperative defense framework	No
Kuo <i>et al.</i> [35]	Edge	Stackelberg	IPS	Resource allocation	No

TABLE 1: Comparison of Related Work

paper, we use the Stackelberg game model to formulate the relationships between MPO and ASPs and solve the subsequent optimization problems to maximize the utility of both sides under DDoS attacks. The optimal IPS deployment is derived given the experienced workload, attack intensity, and efficiency of IPS VMs.

### 3 SYSTEM MODEL

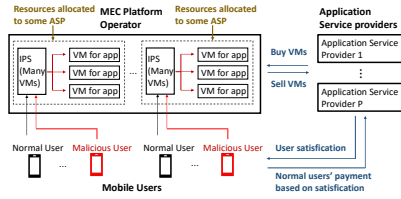


Fig. 1: System Architecture

Notation	Description
$\mathcal{I}$	set of ASPs
$\mathcal{U}_i$	set of total EUs of ASP $i$
$\mathcal{U}_i^n$	set of normal EUs of ASP $i$
$\mathcal{U}_i^m$	set of malicious EUs of ASP $i$
$\lambda_{ij}$	Poisson arrival rate of tasks of EU $j$ of ASP $i$
$\lambda_i^m$	total malicious arrival rate of tasks of MUs of ASP $i$
$\lambda_i^n$	total arrival rate of tasks of NUs of ASP $i$
$d_{ij}$	delay requirements of EU $j$ of ASP $i$
$s_{ij}$	task size of EU $j$ of ASP $i$
$\chi_{ij}$	required CPU cycles of EU $j$ of ASP $i$
$\nu$	the efficiency of IPS VM
$H_i(z_i^h)$	total intercepted malicious requests of ASP $i$
$\Psi_m^v$	price per VM from MPO
$T_{ij}^t$	transmission delay from EU $j$ to ASP $i$
$\mu_i^v$	mean service rate of ASP $i$
$z_i^h$	quantity of deployed IPS VM of ASP $i$
$z_i^v$	quantity of purchased VM of ASP $i$
$a_i, b_i$	lower and upper boundaries of the uniform distribution of delay requirements of ASP $i$
$\xi_i$	minimum ratio of VM reserved for application for ASP $i$
$\gamma_i$	minimum gap between $\mu_i^v$ and $\lambda_i$ of ASP $i$
$\eta_i$	the interception rate of ASP $i$

TABLE 2: Table of Important Notations

We illustrate our system architecture in Figure 1. There are three types of entities in our system: the MPO, ASPs, and EUs. The MPO manages the computational resources in the form of VMs of geographically distributed edge servers. Since the ASPs do not have the computational resources to

process the application tasks, they have intense demands for the MPO's VM, whose total number is  $Q_V$ . The MPO aims to maximize its profit by finding the optimal unit selling price  $\Psi_m^v$  per VM. We denote the set of ASPs as  $\mathcal{I} = \{1, \dots, I\}$ . The ASPs determine the number of VMs to buy from MPO to maximize profit. We use  $z_i^v$  to denote the number of VMs bought by the  $i$ th ASP. Because of the potential network attacks, they utilize these purchased VMs to either run the application service or mitigate the DDoS security attacks. As for the era of B5G/6G, the available network bandwidth is expected to be sufficient to handle massive simultaneous requests. Thus, the main target of DDoS attacks will be the service loading of ASP's server deployed on the VM. This work considers the Application layer DDoS attacks, such as HTTP flooding attacks. When these DDoS attacks occur, the malicious requests flood into ASPs and exhaust the server's computational resources. Because the VM can be used for main service or IPS, there will be a trade-off when deploying VMs. If ASPs devote most of their VMs to running the application service, they are prone to DDoS security attacks. The security threat would damage the users' quality of experience (QoE) or even paralyze the application service such that the revenue of the ASP will be affected. On the other hand, if the ASPs devote most of their VMs to mitigating the DDoS attack, they can not ensure reasonable satisfaction to the users (e.g., users' latency requirements cannot be satisfied) even though the ASPs are immune from the DDoS attacks. Therefore, finding an optimal resource allocation between operating the application service and mitigating the DDoS attacks is crucial.

We denote the set of EUs associated with ASP  $i \in \mathcal{I}$  as  $\mathcal{U}_i$  and further categorize it into two kinds: normal users (NUs)  $\mathcal{U}_i^n$  and malicious users (MUs)  $\mathcal{U}_i^m$ . We assume that the numbers of normal users and malicious users are estimated in advance and remain the same amounts in the considered operation time period. Each EU  $j \in \mathcal{U}_i$  has the Poisson task arrival rate  $\lambda_{ij}$ , latency requirement  $d_{ij}$ , and task size  $s_{ij}$ . We consider that EUs do not have enough computational resources or needed resources such as Graphics Processing Units (GPU) to process the application tasks. Therefore, they will offload the tasks to the MPO. The MUs, on the other hand, will launch attacks by sending malicious requests to the edge server. The DDoS attacks occur when the MUs flood into the edge servers.

The ASPs can intercept some malicious requests by employing IPS such as Suricata [36]. The IPS of all ASPs monitors and filters task requests, and all VMs are devoted to running the application service process and the filtered

task requests. We also assume the IPS does not classify the normal task requests as malicious, but it can not discern all malicious requests. The number of malicious requests the IPS filter depends on the number of VMs the ASPs dedicate to the IPS.

We use  $H(\cdot)$  to represent the relationship between the quantity of intercepted malicious requests and the number of devoted VMs. Without losing generality, we assume that the  $H(\cdot)$  function is non-decreasing and concave. Also,  $H(0) = 0$ . When all malicious requests are intercepted, the ASP cannot intercept more malicious requests by deploying more IPS VMs. Therefore, the value of  $H(x)$  reaches the number of malicious requests and remains this maximal value when  $x$  is larger than a certain value. Inspired by the experiment results in the work [37], where the packet delivery rate is proportional to the CPU utilization percentage. We assume that a VM can filter  $\nu$  packets. Also, we assume that the malicious packets are uniformly distributed. We define the interception rate, which is the number of malicious requests that an IPS VM in the  $i$ th ASP can intercept. We use  $\eta_i$  to denote the interception rate, where  $\eta_i$

$$\eta_i = \frac{\lambda_i^m}{\lambda_i} \times \nu \quad (1)$$

The term  $\lambda_i^m$  indicates the malicious arrival rate of ASP  $i$ , and  $\lambda_i$  indicates the total arrival rate of ASP  $i$ .

In this paper, we use the  $H_i(x)$  function to indicate the relationship between the number of devoted IPS VMs of the  $i$ th ASP and the number of intercepted malicious requests with a linear form of

$$H_i(x) = \min(\eta_i x, \lambda_i^m) \quad (2)$$

We consider an orthogonal frequency-division multiple access (OFDMA) system. The bandwidth of the edge servers whose computational resources are allocated to ASP  $i$  denoted by  $B_i$  can be divided into  $|U_i|$  sub-bands of size, and each sub-band  $W_i = B_i/|U_i|$ . Each EU  $j \in U_i$  transmits the data over each of  $|U_i|$  sub-bands, so they would not cause interference to other EUs while transmitting. The uplink transmission rate  $R_{ij}$  between ASP  $i$  and EU  $j \in U_i$  is

$$R_{ij} = W_i \cdot \log_2 \left( 1 + \frac{p_{ij} g_{ij}}{N_0} \right) \quad (3)$$

where  $p_{ij}$  is the transmission power of EU  $j$  in  $U_i$ ,  $g_{ij}$  denotes the uplink channel gain between ASP  $i$  and EU  $j$ ,  $N_0$  is the background noise power. Based on this rate, we can calculate the mean transmission time of workload consisting of tasks with size  $s_{ij}$  from EU  $j \in U_i$  to ASP  $i$ .

$$T_{ij}^t = \frac{\lambda_{ij} \cdot s_{ij}}{R_{ij}} \quad (4)$$

The task arrival processes at ASP  $i$ , which are offloaded from the EUs, also follow the Poisson process according to the Poisson process's stationary property. Furthermore, the mean arriving rate of the Poisson process at ASP  $i$  can be expressed as  $\lambda_i = \sum_{j \in U_i} \lambda_{ij}$  based on the superposition property of the Poisson process. We use  $\lambda_i^m$  to indicate the total arrival rate of malicious request, which is expressed as  $\lambda_i^m = \sum_{j \in U_i^m} \lambda_{ij}$ . Similarly, we define the total arrival rate of normal request as  $\lambda_i^n$ , where  $\lambda_i^n = \sum_{j \in U_i^n} \lambda_{ij}$ . Since the ASPs buy VMs from MPO, whose overall computational

resources are still limited compared to the cloud platform such as Google Cloud platform or Amazon EC2 platform, we model the meaning processing delay of all VMs purchased by ASPs using  $M/M/1$  queue. Each VM of MPO is homogeneous and can process  $f_m^v$  CPU cycles per second. As the average required CPU cycles for a task of ASP  $i$  may be different, the mean service rate of the queue at ASP  $i$  denoted by  $\mu_i^v$  (the number of tasks that a VM can process on average per second) is different for every ASP  $i$ . We denote the required CPU cycles of user  $j \in U_i$  as  $\chi_{ij}$ . The relationship between  $f_m^v$  and  $\mu_i^v$  is

$$\mu_i^v = f_m^v / \left( \frac{\sum_{j \in U_i} \chi_{ij}}{|U_i|} \right) \quad (5)$$

Furthermore, we let  $\chi_{ij} = B \times s_{ij}$ , where  $B$  is the coefficient of the linear relationship between  $s_{ij}$  and  $\chi_{ij}$ .

If the ASP  $i$  devotes  $z_i^h$  out of  $z_i^v$  VMs to IPS, the intercepted quantity of malicious requests is  $H_i(z_i^h)$ . When ASP  $i$  buys  $z_i^v$  VMs from the MPO and devotes  $z_i^h$  to IPS, those VMs' mean processing delay is

$$T_i^p(z_i^v, z_i^h) = \frac{1}{(z_i^v - z_i^h)\mu_i^v - (\lambda_i - H_i(z_i^h))} \quad (6)$$

where  $(z_i^v - z_i^h)$  is the number of VMs devoted to serving EUs, and  $(z_i^v - z_i^h)\mu_i^v$  is the total service rate of ASP  $i$ . Moreover, the  $H_i(z_i^h)$  is the number of intercepted requests, and  $(\lambda_i - H_i(z_i^h))$  is the number of processed requests.

The payment from the NU  $j \in U_i^n$  to the ASP  $i$  depends on whether the latency requirements  $d_{ij} \forall j \in U_i^n$  are satisfied or not. If latency requirements  $d_{ij}$  are met, NU  $j \in U_i^n$  pays a price  $\Psi_{ij}$  to ASP  $i$ . The heterogeneity of the payment reflects the different characteristics of NUs. The NUs with different latency requirements may have different satisfaction levels with the same service provided by the ASP  $i$ . If latency requirements  $d_{ij}$  are not met, NU  $j \in U_i^n$  does not pay ASP  $i$ . The MUs  $j \in U_i^m$  will not pay ASP  $i$  no matter whether their latency requirement is satisfied or not. We represent the payment of NU  $j \in U_i$  as follows:

$$\mathcal{K}_{ij}(z_i^v, z_i^h) = \begin{cases} \Psi_{ij} & \text{if } T_{ij}^t + T_i^p(z_i^v, z_i^h) \leq d_{ij}, j \in U_i^n \quad (7a) \\ 0 & \text{if } T_{ij}^t + T_i^p(z_i^v, z_i^h) > d_{ij}, j \in U_i^n \quad (7b) \\ 0 & j \in U_i^m \quad (7c) \end{cases}$$

## 4 GAME FORMULATION

Because of the rationalities of both MPO and ASPs, the intrinsic hierarchy between MPO and ASPs, and the influence of one ASP's action on other ASPs' profits, we model the interaction between MPO and ASPs as a two-stage single-leader-multi-followers Stackelberg game. Both the MPO and ASPs are rational players and aim to maximize their profit by choosing their actions. The MPO can select a price per VM  $\Psi_m^v$  that maximizes its profit, and each ASP  $i$  determines the number of VMs to buy from MPO  $z_i^v$  to optimize its profit. The MPO's selection of price per VM affects how many VMs those ASPs would buy, which impacts its revenue. ASPs monitor the actions of the MPO via the price  $\Psi_m^v$ , which the MPO discloses to ASPs, enabling them to determine the quantity of VMs to purchase. Since the ASPs need to buy the computational resources from the MPO to maintain their application services, the MPO has

an advantage over the ASPs. Moreover, due to MPO's finite number of VMs, the more one ASP buys, the less other ASPs can buy. The action of one ASP affects the profit of other ASPs. The Stackelberg game, where the MPO acts as the leader and all ASPs are the followers, can capture such coupled and hierarchical relationships between the MPO and ASPs, the self-interests of both the MPO and ASPs, and the implicit influence among ASPs. Thus, we first illustrate the actions and utilities of MPO and ASPs and then formulate the optimization problems for both players.

#### 4.1 The Action and The Utility of ASPs

Given the MPO's price per VM  $\Psi_m^v$ , each ASP  $i$  chooses the number of VMs to buy from the MPO  $z_i^v$  to maximize its profit, which is the revenue accrued from the normal MU  $j \in U_i^n$  minus the payment to the MPO for purchasing  $z_i^v$  VMs. The natural candidate for the utility is profit. However, even with the same  $z_i^v$ , the revenue of ASP  $i$  hinges on how many VMs it dedicates to the IPS, the distribution of latency requirements  $d_{ij}$ , and the different payments of heterogeneous NU  $j \in U_i^n$ . We define the utility of ASP  $i$  when it purchases  $z_i^v$  VMs from MPO as the maximum expected profit across the different number of VMs dedicated to the IPS  $z_i^h$  with respect to the distribution of latency requirement  $d_{ij}$ . Furthermore, we assume that the latency requirements  $d_{ij} \forall j \in U_i$  follow an uniform distribution indexed by  $i$  over interval  $[a_i, b_i]$  denoted by  $\mathcal{U}(a_i, b_i)$ . To simplify the problem, we relax the variables  $z_i^v, z_i^h \forall i \in \mathcal{I}$  to real numbers. The utility of ASP  $i$  is

$$Y_i(z_i^v | \Psi_m^v) \triangleq \max_{z_i^h} E_{d_{ij}} \left[ \sum_{j \in U_i^n} \mathcal{K}_{ij}(z_i^v, z_i^h) - \Psi_m^v z_i^v \right] \quad (8)$$

$$= \max_{z_i^h} \left[ \sum_{j \in U_i^n} \Psi_{ij} (1 - F_{d_{ij}}(T_{ij}^t + T_i^p(z_i^v, z_i^h))) - \Psi_m^v z_i^v \right] \quad (9)$$

$$\triangleq \max_{z_i^h} X_i(z_i^h | z_i^v, \Psi_m^v) \quad (10)$$

where  $F_X(x)$  is the cumulative distribution function (CDF).

$E_{d_{ij}}$  is derived by taking expectation with respect to the distribution of  $d_{ij}$ , where  $\mathbb{I}\{\cdot\}$  is the indicator function.  $F_{d_{ij}}(T_{ij}^t + T_i^p(z_i^v, z_i^h))$  is the probability that the latency requirement  $d_{ij}$  of the NUs  $j \in U_i^n$  are not satisfied, which is the CDF of  $\mathcal{U}(a_i, b_i)$ . Given  $z_i^v$ , we have three constraints on  $z_i^h$ . First, the number of VMs dedicated to IPS  $z_i^h$  must be no larger than the number of purchased VMs  $z_i^v$ . Second, to stabilize the processing queue, the mean service rate by VMs devoted to running the service must be higher than the mean arrival rate for EUs  $j \in U_i$ . Third, we introduce  $\xi_i \in [0, 1]$  as a resource reservation constraint to control the minimum resource reserved for the service. Thus, to define the utility of ASP  $i$  when the purchased VMs from the MPO are  $z_i^v$ , we must first solve the following optimization problem to decide the optimal configuration of IPS.

$$\begin{aligned} & \text{maximize} && X_i(z_i^h | z_i^v, \Psi_m^v) \\ & z_i^h \in \mathbb{R} \end{aligned} \quad (11a)$$

$$\text{subject to} \quad 0 \leq z_i^h \leq \xi_i z_i^v, \quad (11b)$$

$$0 < (z_i^v - z_i^h) \mu_i^v - \lambda_i + H_i(z_i^h) \quad (11c)$$

The observation that both (11b) and (11c) impose upper bounds on the range of  $z_i^h$  will simplify the characterization of the optimal  $z_i^h$ .

Showing the  $Y_i(z_i^v)$  is well-defined is equivalent to proving the optimization problem (11) has at least one solution. For convenience, we rewrite the term  $\sum_{j \in U_i^n} \Psi_{ij} (1 - F_{d_{ij}}(T_{ij}^t + T_i^p(z_i^v, z_i^h)))$  using the definition of  $F_{d_{ij}}(T_{ij}^t + T_i^p(z_i^v, z_i^h))$ , so it becomes

$$\sum_{j \in U_i^{n1}} \Psi_{ij} \times 1 + \sum_{j \in U_i^{n2}} \Psi_{ij} \times \left( 1 - \frac{T_{ij}^t + T_i^p(z_i^v, z_i^h) - a_i}{b_i - a_i} \right) \quad (12)$$

, where we divide  $U_i^n$  into three sets  $U_i^{n1}$ ,  $U_i^{n2}$  and  $U_i^{n3}$ . We define these three sets as

$$U_i^{n1} = \{T_{ij}^t + T_i^p(z_i^v, z_i^h) < a_i \mid j \in U_i^n\} \quad (13a)$$

$$U_i^{n2} = \{a_i \leq T_{ij}^t + T_i^p(z_i^v, z_i^h) \leq b_i \mid j \in U_i^n\} \quad (13b)$$

$$U_i^{n3} = \{b_i < T_{ij}^t + T_i^p(z_i^v, z_i^h) \mid j \in U_i^n\} \quad (13c)$$

**Lemma 1.** *The utility of ASP  $i$ ,  $Y_i(z_i^v)$ , is well defined, as the optimization problem (11) is a convex optimization problem and has at least one optimal point*

After defining the utility of ASP  $i$ ,  $Y_i(z_i^v)$ , we can now formulate the optimization problem of ASP  $i$  given the MPO's price per VM  $\Psi_m^v$ .

$$\begin{aligned} & \text{maximize} && Y_i(z_i^v | \Psi_m^v) \\ & z_i^v \in \mathbb{R} \end{aligned} \quad (14a)$$

$$\text{subject to} \quad \lambda_i + \gamma_i \leq z_i^v \mu_i^v, \quad (14b)$$

$$0 \leq Y_i(z_i^v | \Psi_m^v) \quad (14c)$$

The constraint (14b) regulates that when the ASP  $i$  dedicates all purchased VMs to running the application service, the mean service rate must be larger than the mean arrival rate by at least  $\gamma_i$ , where  $\gamma_i \geq 0$ . The constraint (14b) also ensures that the feasible region of (11) is non-empty because under this constraint, there is at least one  $z_i^h$  that satisfies (11c). The constraint (14c) represents the individual rationality for every ASP. When making the processing queue stable gives a negative utility, the ASPs would instead choose not to serve the users and not buy any VMs from the MPO, which has zero utility. We defer the analysis of solutions to both optimization problems (11) and (14) to Section 5. We denote the solution to (14) as  $(z_i^v(\Psi_m^v))^*$  to highlight the dependence on the MPO's price per VM  $\Psi_m^v$ .

#### 4.2 The Action and The Utility of The MPO

With the prediction about the number of VMs purchased by ASPs  $(z_i^v(\Psi_m^v))^* \forall i \in \mathcal{I}$ , the MPO chooses the price per VM  $\Psi_m^v$  to maximize its utility which is defined as the MPO's profit. The revenue of the MPO is the sum of payments from all ASPs. Although the ASPs buy those VMs from the MPO, it is the MPO's infrastructure that processes the application requests. As a result, the MPO has an operating cost for keeping the VMs working. We represent the cost for operating  $\sum_{i \in \mathcal{I}} (z_i^v(\Psi_m^v))^*$  VMs by  $C_m^v(\sum_{i \in \mathcal{I}} (z_i^v(\Psi_m^v))^*)$ . The function  $C_m^v(\cdot)$  is assumed to be convex and non-decreasing. We denote the MPO's utility as  $Y_m(\Psi_m^v)$  and formulate the MPO's optimization problem.

$$\begin{aligned} & \text{maximize} && \Psi_m^v \cdot \sum_{i \in \mathcal{I}} (z_i^v(\Psi_m^v))^* - C_m^v \left( \sum_{i \in \mathcal{I}} (z_i^v(\Psi_m^v))^* \right) \\ & \Psi_m^v \end{aligned} \quad (15a)$$

$$\text{subject to } \sum_{i \in \mathcal{I}} (z_i^v(\Psi_m^v))^* \leq Q_v, \quad (15b)$$

$$0 \leq \Psi_m^v \quad (15c)$$

The constraint (15b) mandates the maximum number of VMs  $Q_v$  the MPO can sell. The constraint (15b) imposes a lower bound for MPO's price per VM. The constraint (15c) requires the MPO's price per VM must be non-negative. The property of this optimization (15) relies on the solution  $(z_i^v(\Psi_m^v))^*$ , and we defer the analysis to Section 5.

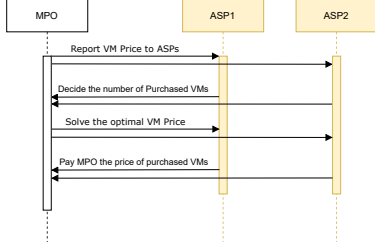


Fig. 2: Flow Chart of the Proposed Stackelberg Game

## 5 GAME OPTIMIZATION

Existing literature presents a range of approaches for determining the Stackelberg equilibrium. One of the most common strategies is to employ Deep Learning and Reinforcement Learning techniques to accelerate the convergence of the equilibrium [38] [39] when determining the exact Stackelberg equilibrium is computationally intractable or no close-form solution is available. In our work, there exists a close-form solution for the followers' (ASPs') best responses. Therefore, we leverage the backward induction principle to derive the Stackelberg equilibrium of the formulated Stackelberg game. That is, we solve the followers' (ASPs') optimization problems first and then the leader's (the MPO's) optimization problem based on the expected best responses of ASPs to the MPO's price per VM, i.e.,  $(z_i^v(\Psi_m^v))^* \forall i \in \mathcal{I}$ .

We formally define the Stackelberg equilibrium [40] of our proposed Stackelberg game. We symbolize the ASP  $i$ 's action set using  $\mathcal{A}_i$  and the profile of all ASPs' action using  $\mathbf{z}^v = (z_1^v, z_2^v, \dots, z_I^v) \in \mathcal{A}_1 \times \mathcal{A}_2 \times \dots \times \mathcal{A}_I$  where  $\mathcal{A}_1 \times \mathcal{A}_2$  means the Cartesian product of  $\mathcal{A}_1$  and  $\mathcal{A}_2$ . The ASP  $i$ 's best response set  $\mathcal{B}_i^R(\Psi_m^v)$  to the MPO's price per VM  $\Psi_m^v$  is

$$\mathcal{B}_i^R(\Psi_m^v) = \{z_i^v \in \mathcal{A}_i | Y_i(z_i^v | \Psi_m^v) \geq Y_i((z_i^v)' | \Psi_m^v) \quad \forall (z_i^v)' \in \mathcal{A}_i, (z_i^v)' \neq z_i^v\} \quad (16)$$

We also denote the Cartesian product of all ASPs' best response sets as

$$\mathcal{B}^R(\Psi_m^v) = \mathcal{B}_1^R(\Psi_m^v) \times \mathcal{B}_2^R(\Psi_m^v) \times \dots \times \mathcal{B}_I^R(\Psi_m^v) \quad (17)$$

Likewise, we denote the MPO's action set as  $\mathcal{A}_m$ . Furthermore, to stress the influence of ASPs' actions on the MPO's utility, we use  $Y_m(\Psi_m^v | \mathbf{z}^v)$  to represent the MPO's utility in the following definition of the MPO's best response set.

$$\mathcal{B}_m = \{\Psi_m^v \in \mathcal{A}_m | Y_m(\Psi_m^v | \mathbf{z}^v) \geq Y_m((\Psi_m^v)' | \mathbf{z}^v) \quad \forall (\Psi_m^v)' \in \mathcal{A}_m, (\Psi_m^v)' \neq \Psi_m^v, \forall \mathbf{z}^v \in \mathcal{B}^R(\Psi_m^v) \quad \forall (\mathbf{z}^v)' \in \mathcal{B}^R((\Psi_m^v)')\} \quad (18)$$

**Definition 1 (Stackelberg equilibrium).** If  $\mathcal{B}_m$  and  $\mathcal{B}^R(\Psi_m^v)$  are both non-empty, the Stackelberg equilibrium in our mechanism is a vector of dimension  $(I + 1)$  that is an element of  $\mathcal{B}_m \times \mathcal{B}^R(\Psi_m^v)$ .

We analyze the process of the formulated Stackelberg game according to whether the solution to (11) is the extreme point or boundary point of the constraint.

### 5.1 ASP Optimization Problem

**Case 1:** The optimal point of (11) is the extreme point whose first derivative with respect to  $z_i^h$ . The first derivative of  $X_i(z_i^h | z_i^v, \Psi_m^v)$  with respect to  $z_i^h$  is

$$[(z_i^v - z_i^h) - \lambda_i + H_i(z_i^h)]^{-2} \cdot (-\mu_i^v + H_i'(z_i^h)) \quad (19)$$

Nevertheless, we can prove that Case 1 will never happen since  $H_i(x) = \min(\eta_i x, \lambda_i^m)$ .

**Lemma 2.** The extreme point of  $X_i(z_i^h | z_i^v, \Psi_m^v)$  doesn't exist in feasible  $z_i^h$  if  $H_i(x) = \min(\eta_i x, \lambda_i^m)$ .

Since the extreme point of (11a) does not exist in feasible  $z_i^h$ , the maximum value of  $X_i(z_i^h | z_i^v, \Psi_m^v)$  only happens at the boundary points of feasible  $z_i^h$ . Moreover, for the linear form of  $H_i(x)$ , the coefficient  $\eta_i$  indicates the number of intercepted malicious requests of an IPS VM. If  $\eta_i$  is larger than the service rate  $\mu_i^v$ , the ASP will tend to deploy as much IPS ratio as possible. In contrast, if  $\eta_i$  is smaller than the service rate  $\mu_i^v$ , the ASP would tend to serve its users instead of intercepting malicious requests.

**Case 2:** In case 2, the interception rate  $\eta_i$  is larger than the service rate  $\mu_i^v$  of the ASP  $i$ . As a result, the ASP has more incentive to deploy as many IPS VMs as possible. In this situation, the optimal point of (11) is the right boundary of (11b) or the exact number of VMs required to defend all of the malicious requests. With a specific form of  $H_i(x) = \min(\eta_i x, \lambda_i^m)$  and the purchased number of VM is  $z_i^v$ , the ASP  $i$ 's optimal number of VMs devoted to the IPS is

$$(z_i^h)^* = \min(\xi_i z_i^v, \frac{\lambda_i^m}{\eta_i}) \quad (20)$$

We substitute  $(z_i^h)^*$  into (6), and the mean processing delay in this case denoted as  $T_{i,2}^p(z_i^v)$  is

$$= \begin{cases} \frac{1}{(1 - \xi_i) z_i^v \mu_i^v - (\lambda_i - \eta_i \xi_i z_i^v)} & z_i^v < \frac{\lambda_i^m}{\eta_i \xi_i} \\ \frac{1}{(z_i^v - \frac{\lambda_i^m}{\eta_i}) \mu_i^v - (\lambda_i - \lambda_i^m)} & z_i^v \geq \frac{\lambda_i^m}{\eta_i \xi_i} \end{cases} \quad (21a) \quad (21b)$$

According to (10), the objective function (14a) becomes

$$Y_i^2(z_i^v | \Psi_m^v) \triangleq \sum_{j \in \mathcal{U}_i^n} \Psi_{ij} (1 - \frac{T_{ij}^t + T_{i,2}^p(z_i^v) - a_i}{b_i - a_i}) - \Psi_m^v z_i^v \quad (22)$$

If  $Y_i^2(z_i^v | \Psi_m^v)$  is concave in feasible  $z_i^v$ , it is much easier to solve the optimization problem (14) with the objective being  $Y_i^2(z_i^v | \Psi_m^v)$ . We prove these properties in the following.

**Lemma 3.**  $Y_i^2(z_i^v | \Psi_m^v)$  is concave in feasible  $z_i^v$ .

The detailed proof is deferred in Appendix C.



With Lemma 3, we can characterize the solution to the optimization problem (14). Before stating the result, we define the feasible region of (14b) as  $\Upsilon_i$  for ASP  $i$  while the feasible region of (14c) as  $\Upsilon_i^2$  when the  $Y_i(z_i^v|\Psi_m^v)$  equals  $Y_i^2(z_i^v|\Psi_m^v)$ .

**Theorem 1.** *The solution to the optimization problem (14) when the objective function (14a) equals  $Y_i^2(z_i^v|\Psi_m^v)$  and  $H_i(x) = \min(\eta_i x, \lambda_i^m)$  is  $(z_{i,2}^{v,e}(\Psi_m^v))^*$  if  $\Upsilon_i \cap \Upsilon_i^2 \neq \emptyset$*

$$= \begin{cases} \frac{\lambda_i + \gamma_i}{\mu_i^v} & (z_{i,2}^{v,e}(\Psi_m^v))^* < \frac{\lambda_i + \gamma_i}{\mu_i^v} \\ (z_{i,2}^{v,e}(\Psi_m^v))^* & (z_{i,2}^{v,e}(\Psi_m^v))^* \geq \frac{\lambda_i + \gamma_i}{\mu_i^v} \end{cases} \quad (23a)$$

$$(23b)$$

where  $(z_{i,2}^{v,e}(\Psi_m^v))^*$

$$= \begin{cases} (z_{i,2}^{v,e''}(\Psi_m^v))^* & \text{if } (z_{i,2}^{v,e'}(\Psi_m^v))^* < \frac{\lambda_i^m}{\eta_i \xi_i} \\ (z_{i,2}^{v,e'}(\Psi_m^v))^* & \text{if } (z_{i,2}^{v,e'}(\Psi_m^v))^* \geq \frac{\lambda_i^m}{\eta_i \xi_i} \end{cases} \quad (24a)$$

$$(24b)$$

where  $(z_{i,2}^{v,e'}(\Psi_m^v))^*$  can be expressed as

$$(z_{i,2}^{v,e'}(\Psi_m^v))^* = \sqrt{\frac{\sum_{j \in \Upsilon_i^n} \Psi_{ij}}{(b_i - a_i) \Psi_m^v \mu_i^v} + \frac{\lambda_i^m}{\eta_i} + \frac{\lambda_i - \lambda_i^m}{\mu_i^v}} \quad (25)$$

and  $(z_{i,2}^{v,e''}(\Psi_m^v))^*$  can be written as

$$(z_{i,2}^{v,e''}(\Psi_m^v))^* = \sqrt{\frac{\sum_{j \in \Upsilon_i^n} \Psi_{ij}}{(b_i - a_i) \Psi_m^v [(1 - \xi_i) \mu_i^v + \xi_i \eta_i]} + \frac{\lambda_i}{(1 - \xi_i) \mu_i^v + \xi_i \eta_i}} \quad (26)$$

and if  $\Upsilon_i \cap \Upsilon_i^2 = \emptyset$

$$(z_{i,2}^{v,e}(\Psi_m^v))^* = 0 \quad (27)$$

The detailed proof is deferred in Appendix D.

From Theorem 1, we can obtain the boundary of  $\Psi_m^i$ , which is the MPO's price per VM  $\Psi_m^i$  that makes ASP  $i$ 's optimal response switch among (27), (23a), and two extreme points, (25) and (26). We can obtain these boundaries by solving  $\Psi_m^v$  that satisfies  $(z_{i,2}^{v,e}(\Psi_m^v))^* = \frac{\lambda_i + \gamma_i}{\mu_i^v}$ ,  $Y_i^2(\frac{\lambda_i + \gamma_i}{\mu_i^v}|\Psi_m^v) = 0$ , and  $(z_{i,2}^{v,e'}(\Psi_m^v))^* = \frac{\lambda_i^m}{\eta_i \xi_i}$ .

**Boundary between (23a) and (26):** If the MPO price  $\Psi_m^v$  satisfies  $(z_{i,2}^{v,e'}(\Psi_m^v))^* < \frac{\lambda_i^m}{\eta_i \xi_i}$ , then  $(z_{i,2}^{v,e}(\Psi_m^v))^* = (z_{i,2}^{v,e''}(\Psi_m^v))^*$ , so the solution to  $(z_{i,2}^{v,e}(\Psi_m^v))^* = \frac{\lambda_i + \gamma_i}{\mu_i^v}$  is

$$\Psi_{m,2}^{i,l''} = \frac{\sum_{j \in \Upsilon_i^n} \Psi_{ij}}{(b_i - a_i)[(1 - \xi_i) \mu_i^v + \xi_i \eta_i]} \times \left[ \frac{\lambda_i + \gamma_i}{\mu_i^v} - \frac{\lambda_i}{(1 - \xi_i) \mu_i^v + \xi_i \eta_i} \right]^{-2} \quad (28)$$

**Boundary between (23a) and (25):** If the MPO price  $\Psi_m^v$  satisfies  $(z_{i,2}^{v,e'}(\Psi_m^v))^* \geq \frac{\lambda_i^m}{\eta_i \xi_i}$ , then  $(z_{i,2}^{v,e}(\Psi_m^v))^* = (z_{i,2}^{v,e'}(\Psi_m^v))^*$ , so the solution to  $(z_{i,2}^{v,e}(\Psi_m^v))^* = \frac{\lambda_i + \gamma_i}{\mu_i^v}$  is

$$\Psi_{m,2}^{i,l'} = \frac{\sum_{j \in \Upsilon_i^n} \Psi_{ij}}{(b_i - a_i)} \times \left[ \frac{\lambda_i^m + \gamma_i}{\mu_i^v} - \frac{\lambda_i^m}{\eta_i} \right]^{-2} \quad (29)$$

**Boundary between (23a) and (27):** The solution to the equation  $Y_i^2(\frac{\lambda_i + \gamma_i}{\mu_i^v}|\Psi_m^v) = 0$  is

$$\Psi_{m,2}^{i,z} = \sum_{j \in \Upsilon_i^n} \Psi_{ij} \left( 1 - \frac{T_{ij}^t + T_{i,2}^p(\frac{\lambda_i + \gamma_i}{\mu_i^v}) - a_i}{b_i - a_i} \right) \times \left( \frac{\mu_i^v}{\lambda_i + \gamma_i} \right) \quad (30)$$

**Boundary between (26) and (27):** The utility of ASP may descend to 0 when increasing the price even before reaching the queuing stability constraint. Therefore, we need to solve another equation  $Y_i^2((z_{i,2}^{v,e}(\Psi_m^v))^*|\Psi_m^v) = 0$  to find the  $\Psi_m^v$  which makes the ASP  $i$ 's optimal response switch among (23b) and 0. If the MPO price  $\Psi_m^v$  satisfies  $(z_{i,2}^{v,e'}(\Psi_m^v))^* < \frac{\lambda_i^m}{\eta_i \xi_i}$ , the best response  $(z_{i,2}^{v,e}(\Psi_m^v))^* = (z_{i,2}^{v,e''}(\Psi_m^v))^*$ , and the solution is

$$\Psi_{m,2}^{i,r''} = \frac{\sum_{j \in \Upsilon_i^n} \Psi_{ij} \frac{b_i - T_{ij}^t}{b_i - a_i} \lambda_i + 2 \frac{\sum_{j \in \Upsilon_i^n} \Psi_{ij}}{b_i - a_i}}{[(1 - \xi_i) \mu_i^v + \xi_i \eta_i] \left[ \frac{\sum_{j \in \Upsilon_i^n} \Psi_{ij} \frac{b_i - T_{ij}^t}{b_i - a_i} \lambda_i + 2 \frac{\sum_{j \in \Upsilon_i^n} \Psi_{ij}}{b_i - a_i}}{\lambda_i^2} - 2 \sqrt{\left( \frac{\sum_{j \in \Upsilon_i^n} \Psi_{ij}}{b_i - a_i} \right)^2 + \sum_{j \in \Upsilon_i^n} \Psi_{ij} \frac{b_i - T_{ij}^t}{b_i - a_i} \frac{\sum_{j \in \Upsilon_i^n} \Psi_{ij}}{b_i - a_i} \lambda_i} \right]} \quad (31)$$

**Boundary between (25) and (27):** If the MPO price  $\Psi_m^v$  satisfies  $(z_{i,2}^{v,e}(\Psi_m^v))^* \geq \frac{\lambda_i^m}{\eta_i \xi_i}$ ,  $(z_{i,2}^{v,e}(\Psi_m^v))^* = (z_{i,2}^{v,e'}(\Psi_m^v))^*$ , and the result  $\Psi_{m,2}^{i,r'}$  is shown below

$$\Psi_{m,2}^{i,r'} = \frac{\sum_{j \in \Upsilon_i^n} \Psi_{ij} \frac{b_i - T_{ij}^t}{b_i - a_i} \mu_i \left[ \frac{\lambda_i^m}{\eta_i} \mu_i^v + \lambda_i - \lambda_i^m \right] + 2 \frac{\sum_{j \in \Upsilon_i^n} \Psi_{ij}}{b_i - a_i} \mu_i}{\left[ \frac{\lambda_i^m}{\eta_i} \mu_i^v + \lambda_i - \lambda_i^m \right]^2} - \frac{2 \mu_i^v \sqrt{\left( \frac{\sum_{j \in \Upsilon_i^n} \Psi_{ij}}{b_i - a_i} \right)^2 + \sum_{j \in \Upsilon_i^n} \Psi_{ij} \frac{b_i - T_{ij}^t}{b_i - a_i} \frac{\sum_{j \in \Upsilon_i^n} \Psi_{ij}}{b_i - a_i} \left[ \frac{\lambda_i^m}{\eta_i} \mu_i^v + \lambda_i - \lambda_i^m \right]}}{\left[ \frac{\lambda_i^m}{\eta_i} \mu_i^v + \lambda_i - \lambda_i^m \right]^2} \quad (32)$$

**Boundary between (25) and (26):** We can also obtain the MPO's price that the ASP's response changes from (24a) to (24b) by solving the  $\Psi_m^v$  that satisfied  $(z_{i,2}^{v,e'}(\Psi_m^v))^* = \frac{\lambda_i^m}{\eta_i \xi_i}$ .

$$\Psi_{m,2}^{i,c} = \left( \frac{\sum_{j \in \Upsilon_i^n} \Psi_{ij}}{(b_i - a_i) \mu_i^v} \right) \times \left( \frac{\lambda_i^m}{\eta_i \xi_i} - \frac{\lambda_i - \lambda_i^m}{\mu_i^v} - \frac{\lambda_i^m}{\eta_i} \right)^{-2} \quad (33)$$

By the boundaries we have obtained above, we can precisely predict the  $i$ th ASP's best response  $z_i^v$  given any  $\Psi_m^v$ . There are four possible outcomes.

If  $\Psi_{m,2}^{i,z} > \Psi_{m,2}^{i,l'}$  and  $\frac{\lambda_i + \gamma_i}{\mu_i^v} > \frac{\lambda_i^m}{\eta_i \xi_i}$  then the ASP is classified as **Case A**.

$$z_i^v = \begin{cases} 0 & \Psi_m^v \in (\Psi_{m,2}^{i,z}, \infty) \end{cases} \quad (34a)$$

$$\frac{\lambda_i + \gamma_i}{\mu_i^v} \quad \Psi_m^v \in [\Psi_{m,2}^{i,l'}, \Psi_{m,2}^{i,z}] \quad (34b)$$

$$(z_{i,2}^{v,e'}(\Psi_m^v))^* \quad \Psi_m^v \in [0, \Psi_{m,2}^{i,l'}) \quad (34c)$$

Moreover, if  $(z_{i,2}^{v,e'}(\Psi_{m,2}^{i,r'}))^* > \frac{\lambda_i^m}{\eta_i \xi_i}$  and  $(z_{i,2}^{v,e'}(\Psi_{m,2}^{i,r'}))^* > \frac{\lambda_i + \gamma_i}{\mu_i^v}$  the ASP is classified as **Case B**.

$$z_i^v = \begin{cases} 0 & \Psi_m^v \in [\Psi_{m,2}^{i,r'}, \infty) \\ (z_{i,2}^{v,e'}(\Psi_m^v))^* & \Psi_m^v \in [0, \Psi_{m,2}^{i,r'}) \end{cases} \quad (35a)$$

$$(35b)$$

In addition, if  $\frac{\lambda_i}{\eta_i \xi_i} > \frac{\lambda_i + \gamma_i}{\mu_i^v}$  and  $\Psi_{m,2}^{i,z} < \Psi_{m,2}^{i,l''}$ , the ASP is classified as **Case C**.

$$z_i^v = \begin{cases} 0 & \Psi_m^v \in (\Psi_{m,2}^{i,z}, \infty) \\ \frac{\lambda_i + \gamma_i}{\mu_i^v} & \Psi_m^v \in [\Psi_{m,2}^{i,l''}, \Psi_{m,2}^{i,z}] \\ (z_{i,2}^{v,e''}(\Psi_m^v))^* & \Psi_m^v \in [\Psi_{m,2}^{i,c}, \Psi_{m,2}^{i,l''}] \\ (z_{i,2}^{v,e'}(\Psi_m^v))^* & \Psi_m^v \in [0, \Psi_{m,2}^{i,c}) \end{cases} \quad (36a)$$

$$(36b)$$

$$(36c)$$

$$(36d)$$

If the ASP does not meet the conditions above, the ASP is classified as **Case D**.

$$z_i^v = \begin{cases} 0 & \Psi_m^v \in [\Psi_{m,2}^{i,r''}, \infty) \\ (z_{i,2}^{v,e''}(\Psi_m^v))^* & \Psi_m^v \in [\Psi_{m,2}^{i,c}, \Psi_{m,2}^{i,r''}] \\ (z_{i,2}^{v,e'}(\Psi_m^v))^* & \Psi_m^v \in [0, \Psi_{m,2}^{i,c}) \end{cases} \quad (37a)$$

$$(37b)$$

$$(37c)$$

Note that the four cases mentioned above cover all the possible cases. To elaborate, the extreme point (24) has two possibilities: (24a) and (24b) exists or only (24b) exists. In addition, (23) has two possibilities: the extreme point (24) is always higher than the queuing stability constraint, or the extreme point (24) is lower than the queuing stability constraint in some given price of VM  $\Psi_m^v$ . As a result, there are  $2 \times 2 = 4$  possibilities. To conclude, the MPO can anticipate the response of the ASP given an arbitrary price of VM  $\Psi_m^v$ .

**Case 3:** Case 3 is the case that the interception rate  $\eta_i$  is smaller than the  $i$ th ASP's service rate  $\mu_i^v$ . In this case, the optimal point of (11) occurs at the left boundary of (11b). That is, when the purchased number of VM is  $z_i^v$ , the ASP  $i$ 's optimal number of VMs devoted to the IPS  $(z_i^h)^*$  is

$$(z_i^h)^* = 0 \quad (38)$$

When we substitute (38) into (6), the mean processing delay denoted as  $T_{i,3}^p(z_i^v)$  depends only on  $z_i^v$  and is

$$T_{i,3}^p(z_i^v) = \frac{1}{z_i^v \mu_i^v - \lambda_i} \quad (39)$$

According to (10), the objective function (14a) becomes

$$Y_i^3(z_i^v | \Psi_m^v) \triangleq \sum_{j \in U_i^n} \Psi_{ij} \left(1 - \frac{T_{ij}^t + T_{i,3}^p(z_i^v) - a_i}{b_i - a_i}\right) - \Psi_m^v z_i^v \quad (40)$$

If  $Y_i^3(z_i^v | \Psi_m^v)$  is concave in feasible  $z_i^v$ , it is much easier to solve the optimization problem (14) with the objective being  $Y_i^3(z_i^v | \Psi_m^v)$ . We prove this properties in the following.

**Lemma 4.**  $Y_i^3(z_i^v | \Psi_m^v)$  is concave in feasible  $z_i^v$ .

With Lemma 4, we can characterize the solution to the optimization problem (14) when the objective function (14a) equals  $Y_i^3(z_i^v | \Psi_m^v)$  in the following theorem with a specific form of  $H_i(\cdot)$ . We define the feasible region of (14c) as  $\Upsilon_i^3$  when the  $Y_i(z_i^v | \Psi_m^v)$  equals  $Y_i^3(z_i^v | \Psi_m^v)$ .

**Theorem 2.** The solution to the optimization problem (14) when the objective function (14a) equals  $Y_i^3(z_i^v | \Psi_m^v)$  and  $H_i(x) = \min(\eta_i x, \lambda_i^m)$  is  $(z_{i,3}^v(\Psi_m^v))^*$  and if  $\Upsilon_i \cap \Upsilon_i^3 \neq \emptyset$

$$= \begin{cases} \frac{\lambda_i + \gamma_i}{\mu_i^v} & (z_{i,3}^v(\Psi_m^v))^* < \frac{\lambda_i + \gamma_i}{\mu_i^v} \\ (z_{i,3}^v(\Psi_m^v))^* & (z_{i,3}^v(\Psi_m^v))^* \geq \frac{\lambda_i + \gamma_i}{\mu_i^v} \end{cases} \quad (41a)$$

$$(41b)$$

where

$$(z_{i,3}^v(\Psi_m^v))^* = \sqrt{\frac{\sum_{j \in U_i^n} \Psi_{ij}}{(b_i - a_i) \Psi_m^v \mu_i^v}} + \frac{\lambda_i}{\mu_i^v} \quad (42)$$

and if  $\Upsilon_i \cap \Upsilon_i^3 = \emptyset$

$$(z_{i,3}^v(\Psi_m^v))^* = 0 \quad (43)$$

From Theorem 2, we can obtain the boundary of  $\Psi_m^i$ , which is the MPO's price per VM  $\Psi_m^i$  that makes ASP  $i$ 's optimal response switch among (43), (41a) and (41b) by solving the  $\Psi_m^v$  that satisfies  $(z_{i,3}^v(\Psi_m^v))^* = \frac{\lambda_i + \gamma_i}{\mu_i^v}$  and  $Y_i^3(\frac{\lambda_i + \gamma_i}{\mu_i^v} | \Psi_m^v) = 0$ .

**Boundary between (41a) and (42), (41a) and (43):** The boundaries can be expressed as

$$\Psi_{m,3}^{i,l} = \frac{\sum_{j \in U_i^n} \Psi_{ij}}{(b_i - a_i)} \times \left[ \frac{\lambda_i + \gamma_i}{\mu_i^v} - \frac{\lambda_i}{\mu_i^v} \right]^{-2} \quad (44)$$

$$\Psi_{m,3}^{i,z} = \sum_{j \in U_i^n} \Psi_{ij} \left(1 - \frac{T_{ij}^t + T_{i,3}^p(\frac{\lambda_i + \gamma_i}{\mu_i^v}) - a_i}{b_i - a_i}\right) \times \left(\frac{\mu_i^v}{\lambda_i + \gamma_i}\right) \quad (45)$$

**Boundary between (42) and (43):** Regardless, the ASP's utility may descend to 0 without experiencing the state of maintaining the queuing stability. Therefore, we need to solve the equation  $Y_i^3((z_{i,3}^v(\Psi_m^v))^*) = 0$  to find the  $\Psi_m^i$  which makes the ASP  $i$ 's optimal response switch among (41b) and 0.

$$\Psi_{m,3}^{i,r} = \frac{\sum_{j \in U_i^n} \Psi_{ij} \frac{b_i - T_{ij}^t}{b_i - a_i} \mu_i \lambda_i + 2 \frac{\sum_{j \in U_i^n} \Psi_{ij}}{b_i - a_i} \mu_i}{\lambda_i^2} - \frac{2 \mu_i \sqrt{\left(\frac{\sum_{j \in U_i^n} \Psi_{ij}}{b_i - a_i}\right)^2 + \sum_{j \in U_i^n} \Psi_{ij} \frac{b_i - T_{ij}^t}{b_i - a_i} \frac{\sum_{j \in U_i^n} \Psi_{ij}}{b_i - a_i} \lambda_i}}{\lambda_i^2} \quad (46)$$

By the boundaries we have obtained above, we can precisely predict the  $z_i^v$  given any  $\Psi_m^v$ . We can divide them into two cases. One is  $\Psi_{m,3}^{i,z} \geq \Psi_{m,3}^{i,l}$ , the other is  $\Psi_{m,3}^{i,z} < \Psi_{m,3}^{i,l}$ . If  $\Psi_{m,3}^{i,z} > \Psi_{m,3}^{i,l}$ , the extreme point (41b) will be lower than the queuing stability constraint as some given  $\Psi_m^v$ .

$$z_i^v = \begin{cases} 0 & \Psi_m^v \in (\Psi_{m,3}^{i,z}, \inf) \\ \frac{\lambda_i + \gamma_i}{\mu_i^v} & \Psi_m^v \in [\Psi_{m,3}^{i,l}, \Psi_{m,3}^{i,z}] \end{cases} \quad (47a)$$

$$(47b)$$

$$(47c)$$

If  $\Psi_{m,3}^{i,z} \leq \Psi_{m,3}^{i,l}$ , the extreme point (41b) is always higher than the queuing stability constraint

$$z_i^v = \begin{cases} 0 & \Psi_m^v \in [\Psi_{m,3}^{i,r}, \inf) \\ (z_{i,3}^v(\Psi_m^v))^* & \Psi_m^v \in [0, \Psi_{m,3}^{i,r}) \end{cases} \quad (48a)$$

$$(48b)$$



Based on the analysis above, the MPO can predict the number of VMs every ASP will buy given an arbitrary price per VM in each case. Thus, the MPO can optimize its utility by setting  $\Psi_m^v$  accordingly. The corresponding optimization problem is in the following subsection.

## 5.2 MPO Optimization Problem

We solve the optimization problem (15a) in this subsection. For every ASP, the optimal solution is either Case 2 or Case 3. That is,  $\mathcal{I}_{case2} \cup \mathcal{I}_{case3} = \mathcal{I}$ , where  $\mathcal{I}_{case2}$  indicates the ASP set which its optimal solution is in Case 2, and  $\mathcal{I}_{case3}$  indicates the ASP set which its optimal solution is in Case 3. The boundary set of an ASP is  $L_i^b$ , and because of the different characteristics of ASPs, there are six cases.

$$L_i^b = \begin{cases} \{\Psi_{m,2}^{i,z}, \Psi_{m,2}^{i,l'} \mid i \in \mathcal{I}_{case2}\} & (49a) \\ \{\Psi_{m,2}^{i,z}, \Psi_{m,2}^{i,l'}, \Psi_{m,2}^{i,c} \mid i \in \mathcal{I}_{case2}\} & (49b) \\ \{\Psi_{m,2}^{i,r'} \mid i \in \mathcal{I}_{case2}\} & (49c) \\ \{\Psi_{m,2}^{i,r''}, \Psi_{m,2}^{i,c} \mid i \in \mathcal{I}_{case2}\} & (49d) \\ \{\Psi_{m,3}^{i,z}, \Psi_{m,3}^{i,l} \mid i \in \mathcal{I}_{case3}\} & (49e) \\ \{\Psi_{m,3}^{i,r} \mid i \in \mathcal{I}_{case3}\} & (49f) \end{cases}$$

We use  $L^b$  to indicate the boundary sequence of all ASPs, where  $L^b = \bigcup_{i=1}^{|\mathcal{I}|} L_i^b$ .

By analyzing the boundary of MPO's price in each case, we find that the space of MPO's price per VM can be divided into several intervals based on the different  $z_i^v$ . Therefore, the boundary sequence partitions the space of the MPO's price per VM  $\Psi_m^v$  into  $J = |L^b| + 1$  intervals.

$$0 \cup \mathbb{R}^+ = \mathcal{M}^1 \cup \mathcal{M}^2 \dots \cup \mathcal{M}^J \quad (50)$$

Within each interval  $\mathcal{M}^k, \forall k \in \{1, \dots, J\}$ , we define five sets of ASP  $i$  with different optimal responses.

$$\begin{aligned} \Omega^{k,z} &= \{i \in \mathcal{I} \mid (z_i^v(\Psi_m^v))^* = 0\} \\ \Omega^{k,l} &= \{i \in \mathcal{I} \mid (z_i^v(\Psi_m^v))^* = \frac{\lambda_i + \gamma_i}{\mu_i^v}\} \\ \Omega_2^{k,e'} &= \{i \in \mathcal{I}_{case2} \mid (z_i^v(\Psi_m^v))^* = (z_{i,2}^{v,e'}(\Psi_m^v))^*\} \\ \Omega_2^{k,e''} &= \{i \in \mathcal{I}_{case2} \mid (z_i^v(\Psi_m^v))^* = (z_{i,2}^{v,e''}(\Psi_m^v))^*\} \\ \Omega_3^{k,e} &= \{i \in \mathcal{I}_{case3} \mid (z_i^v(\Psi_m^v))^* = (z_{i,3}^{v,e}(\Psi_m^v))^*\} \end{aligned} \quad (51)$$

The union of these five sets includes all ASPs. That is,  $\mathcal{I} = \Omega^{k,z} \cup \Omega^{k,l} \cup \Omega_2^{k,e'} \cup \Omega_2^{k,e''} \cup \Omega_3^{k,e}, \forall k \in \{1, \dots, J\}$ . The total number of VMs bought by the ASPs within each interval  $\mathcal{M}^k, \forall k \in \{1, \dots, J\}$  can be expressed

$$\begin{aligned} \sum_{i \in \mathcal{I}} (z_i^v(\Psi_m^v))^* &= \sum_{i \in \Omega^{k,l}} \frac{\lambda_i + \gamma_i}{\mu_i^v} + \sum_{i \in \Omega_2^{k,e'}} (z_{i,2}^{v,e'}(\Psi_m^v))^* + \\ &\quad \sum_{i \in \Omega_2^{k,e''}} (z_{i,2}^{v,e''}(\Psi_m^v))^* + \sum_{i \in \Omega_3^{k,e}} (z_{i,3}^{v,e}(\Psi_m^v))^* \end{aligned} \quad (52)$$

With (52), the MPO's optimization problem in each interval  $\mathcal{M}^k, \forall k \in \{1, \dots, J\}$  is

$$\begin{aligned} \text{maximize}_{\Psi_m^v} \quad & \Psi_m^v \cdot \sum_{i \in \mathcal{I}} (z_i^v(\Psi_m^v))^* - C_m^v \left( \sum_{i \in \mathcal{I}} (z_i^v(\Psi_m^v))^* \right) \end{aligned} \quad (53a)$$

$$\text{subject to} \quad \sum_{i \in \mathcal{I}} (z_i^v(\Psi_m^v))^* \leq \mathcal{Q}_v, \quad (53b)$$

$$\Psi_m^v \in \mathcal{M}^k \quad (53c)$$

**Theorem 3.** In each interval  $\mathcal{M}^k, \forall k \in \{1, \dots, J\}$ , the MPO's optimization problem (53) is a convex optimization problem. The MPO's optimal price per VM  $\Psi_m^v$  is one of the optimal prices in each  $\mathcal{M}^k, \forall k \in \{1, \dots, J\}$  that offers the highest utility.

As the optimization problem (53) is convex in every interval, the global optimal solution  $\Psi_m^v$  for MPO can be identified efficiently by comparing the optimal value in every interval.

After analyzing both the followers' and the leader's optimization problems, we now prove that there exists at least one Stackelberg game equilibrium in our system with minor modifications. We convert all the right open intervals among  $\mathcal{M}^k, \forall k \in \{1, \dots, J\}$  into right close intervals with the right endpoint being the right endpoint of original ones minus small constants  $\epsilon$ , so all the intervals  $\mathcal{M}^k, \forall k \in \{1, \dots, J\}$  are compact sets.

**Theorem 4.** There exists a Stackelberg equilibrium in our system after the modifications.

Because of Theorem 3, if the specific form of  $C_m^v(\cdot)$  is given, we can solve the exact form of the solution to the MPO's optimization problem. We will select some specific forms for  $C_m^v(\cdot)$  in the Section 6.

## 5.3 Complexity analysis

We analyze the time complexity of deriving Stackelberg game. At first, we calculate the boundaries of the MPO price  $\Psi_m^v$  for every best response for ASPs, which has linear time complexity  $O(|\mathcal{I}|)$ . Later, we calculate the optimal price for MPO for every interval using a convex optimization solver. Let  $f(\cdot)$  be the time complexity of the convex optimization solver, we can derive the optimal price in every interval in  $O(J \cdot f(\cdot))$ . Lastly, we compare the optimal price for every interval, which can be derived in  $O(J)$ . Therefore, the overall time complexity is  $O(J \cdot f(\cdot) + J + |\mathcal{I}|) = O(J \cdot f(\cdot))$ .

## 6 SIMULATION RESULTS

In this simulation, we evaluate the effectiveness of the proposed Stackelberg equilibrium by conducting simulations on a MEC system consisting of one MPO and five ASPs, as discussed in [41]. The CPU frequency per VM ( $f_m^v$ ) for the MPO is set to 0.2GHz, and the cost function is defined as  $C_m^v(x) = 0.01x^2$ . The frequency of a MEC server is 20GHz, and we assume that there are ten servers in the MEC platform, resulting in a total of 1000 VMs for an MPO. For each ASP, we set the bandwidth ( $B_i$ ) to 0.4GHz, and the total bandwidth in the system is 2GHz. The transmission power ( $p_i$ ) is set to 23dBm [42], and the background noise power ( $N_0$ ) to -100dBm [43]. We use the path loss model  $g(d) = 22\log_{10}(d) + 28 + 20\log_{10}(f_c)$  [44], where  $d$  denotes the distance between EUs and ASPs, and  $f_c$  is the frequency band. Other system parameters include  $\xi_i$ , which is chosen uniformly between  $[0.1, 0.2]$ ,  $\gamma_i$ , which is chosen uniformly between  $[20, 40]$ , and the frequency band  $f_c$ , which is set to 2.1GHz. For EUs, we assume that the distance between the

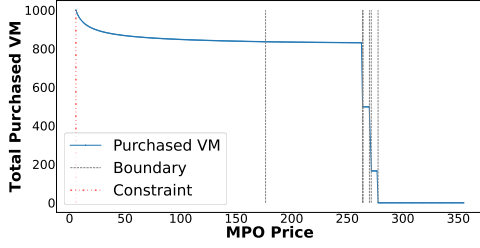


Fig. 3: Total Purchased VM v.s. MPO Price

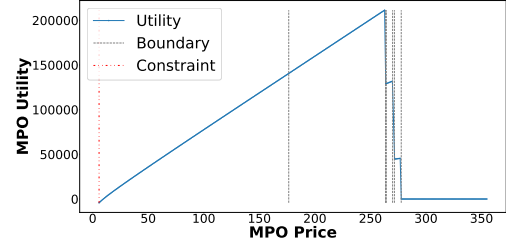


Fig. 4: MPO Utility v.s. MPO Price

EU and ASP is 50m [45]. The task size ( $s_{ij}$ ) of each EU  $j$  is set to an average of 10KB [46], where the size of every task is uniformly chosen from 8KB to 12KB. The latency requirements of the NU are set uniformly between 15ms to 1000ms, and the required CPU cycle per bit ( $B$ ) is set to 500 [46]. The arrival rate ( $\lambda_{ij}$ ) is chosen uniformly between 0.6 and 1 [47]. The payments ( $\Psi_{ij}$ ) are chosen uniformly between 1 and 100. If not specified, the ratio of MUs to total EUs is 0.1, the number of EUs is 1000 per ASP, and the IPS efficiency value, the number of packets that a VM can filter,  $\nu$  is set to 50. To solve the convex optimization problem, we use CVXPY [48] [49] as our convex solver and the Embedded Conic Solver (ECOS) [50]. We set the constant  $\epsilon$  to  $10^{-4}$ .

In the simulations, we evaluate six different schemes: the **Proposed Scheme** (Proposed), **No IPS**, **5% IPS VM** (5% IPS), **7% IPS VM** (7% IPS), **10% IPS VM** (10% IPS), **Proportional Malicious Ratio** (Prop Malicious ratio), and **Proportional IPS Efficiency** (Prop IPS efficiency). In the *Proposed Scheme*, the ASPs can optimize their utility by purchasing VMs and allocating their IPS VMs. At the same time, the MPO determines the price optimally, and the system achieves the proposed Stackelberg equilibrium. In the *No IPS* scheme, the ASPs do not allocate VMs for the IPS services. Instead, the ASPs devote all of their VMs to serving EUs. For the fixed ratio cases, i.e., *5% IPS VM*, *7% IPS VM*, and *10% IPS VM*, the ASPs allocate a fixed ratio of VMs for IPS services. In the *Proportional Malicious Ratio* scheme, the ASP devotes VMs to IPS services based on the ratio of malicious to total users. In this simulation, the ASPs devoted  $30 \times r\%$  VMs to IPS services, where  $r$  is the ratio of malicious to total users. As for the *Proportional IPS Efficiency* scheme, the ASPs devote VMs to IPS services based on the efficiency of IPS  $\nu$ , which is the number of packets an IPS VM can filter. We set the ratio of IPS VMs to  $0.01 \times \nu\%$ , where  $\nu$  is the IPS efficiency.

## 6.1 MPO Optimization

In this simulation, we verify the convexity of the MPO optimization problem. The boundaries of price interval have been plotted in Figure 3 and Figure 4. The ASPs have the same optimal response in every interval. As the MPO price increases, the optimal responses by the ASPs change from a decreasing, convex function (24) and (42) to a constant. Additionally, the MPO utility shown in Figure 4 is a concave function in every interval.

## 6.2 Comparison Between Various Number of EUs

In this section, we compare multiple IPS VM deployment schemes on ASPs with various numbers of EUs. We consider two cases: ASP with high workload EUs and ASP with low workload EUs. In ASP with high and low workload EUs, the task size  $s_{ij}$  is chosen uniformly between 11 – 12 kB and 8 – 9 kB, respectively. We discuss the results under these two types of ASP in the following.

**ASP with high workload EUs:** When considering ASP with high workload EUs, by (5), we observe that the service rate is lower than the low workload case due to the increased required CPU cycles. In this simulation, the service rate is lower than the interception rate of IPS VMs. Thus, the ASP is incentivized to deploy VMs on IPS services as much as possible to eradicate malicious requests. In Figure 5b, the 7% IPS VM scheme has better utility than 5% IPS VM scheme because ASP has the incentive to deploy IPS services, moreover, in Figure 5c and Figure 5a we can discover the same phenomenon. However, 10% IPS VM has lower utility due to the deployment of redundant IPS. When the number of users increases, the utility of ASPs starts to increase because they can serve more users, and the utility of MPO increases because the ASPs buy more VMs from the MPO to serve more users, so the total social welfare also increases. In this scenario, both *Proportional Malicious Ratio* and *Proportional IPS Efficiency* allocate a fixed ratio of IPS VMs because the ratio of malicious to total users and IPS efficiency are both fixed, which is  $30 \times 0.1 = 3\%$  and  $0.01 \times 50 = 0.5\%$ , respectively. Therefore, the utility curves of both schemes in Figure 5b are similar to the fixed IPS scheme (5% IPS VM, 7% IPS VM, and 10% IPS VM). In addition, our *Proposed Scheme* has outperformed other schemes in Figure 5a, Figure 5b, and Figure 5c.

**ASP with low workload EUs:** Considering ASP with low workload EUs, the required CPU cycles are low. Unlike the high-workload case, in this case, the service rate is higher than the interception rate of IPS VMs, and the ASP has more incentive to deploy VMs to serve EUs than to wipe out malicious requests. Therefore, the optimal deployment scheme for such ASPs is identical to the *No IPS* scheme. We can observe the phenomenon in Figure 6b. Moreover, the ASPs' utility in 5% IPS VM, 7% IPS VM, and 10% IPS VM are lower because it is less efficient for ASPs to deploy IPS VMs than to deploy VMs to serve users. Similar to the high-workload case, both schemes *Proportional Malicious Ratio* and *Proportional IPS Efficiency* allocate a fixed ratio of IPS VMs. In this scenario, the *Proportional Malicious Ratio* scheme allocates  $30 \times 0.1 = 3\%$  of IPS VMs, and the

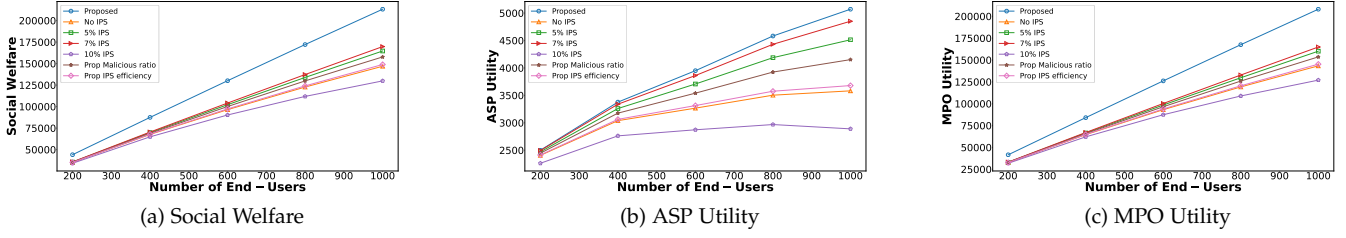


Fig. 5: Different Number of High Workload EUs

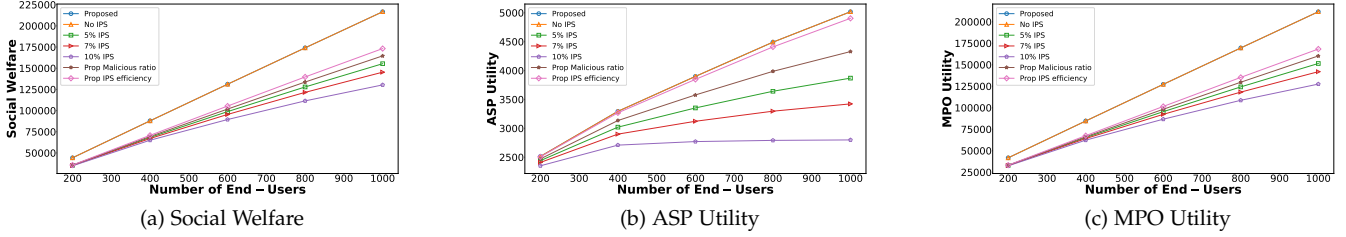


Fig. 6: Different Number of Low Workload EUs

*Proportional IPS Efficiency* scheme allocates  $0.01 \times 50 = 0.5\%$  of IPS VMs. As a result, in Figure 6b, the *Proportional Malicious Ratio* has lower utility than *Proportional IPS Efficiency* because it is less efficient to deploy IPS VMs in this scenario. In addition, our proposed scheme has the highest utility in Figure 6a, Figure 6b, and Figure 6c.

### 6.3 Comparison Between Various Malicious Users to Total Users Ratio

We compare different IPS VM deployment schemes on ASPs given various malicious users to total users ratios. Similar to the previous simulation, we discuss two different cases: ASP with high workload EUs and ASP with low workload EUs. The parameters of high and low workload EUs are mentioned in the previous simulation.

**ASP with high workload EUs:** When EUs' workload are high, by Equation (5), the service rate would be low when applying default parameters (ratio = 0.1). Therefore, the interception rate would be higher than the service rate, and the ASPs tend to deploy more IPS VMs. However, when the number of malicious users is low, the IPS VMs can intercept fewer malicious requests, and the interception rate would be lower than the service rate. As the number of malicious users increases, the behavior of the ASPs changes at a specific point, which is the point where the interception rate and the service rate intersect.

When the number of malicious users are low, the ASP will tend to deploy fewer IPS VM. In this case, if the ASP deploys excessive IPS VMs, it would crowd out VMs serving EUs. The scheme *No IPS* has higher utility when the number of malicious users are low. As for *10% IPS VM*, the utility curve in Figure 7b is far lower than *5% IPS VM* because it deploys redundant IPS VMs. Moreover, when the ratio of malicious to total users increases, the interception rate increases and exceed the service rate. Therefore, the ASP tends to deploy more IPS VMs. We observe that the

curve of *No IPS* scheme and the utility curve of *5% IPS VM* and *7% IPS VM* intersects. Also, when the malicious ratio increases, the *10% IPS VM* scheme has better utility than the *5% IPS VM* scheme.

For the scheme *Proportional Malicious Ratio*, when the malicious ratio is low, *Proportional Malicious Ratio* deploys fewer IPS VMs, so the utility is lower than *5% IPS VM*. However, when the number of malicious users increases, the *Proportional Malicious Ratio* deploys more IPS VMs than *5% IPS VM* and *10% IPS VM*, and the ASPs' utility is higher than the scheme mentioned above. In addition, the *Proposed Scheme* outperformed other schemes in Figure 7a, Figure 7b, and Figure 7c due to the optimal IPS deployment.

**ASP with low workload EUs:** In this case, when the workload of the EUs are low, by Equation (5), the service rate would be high in default parameters (ratio = 0.1). Also, the interception rate is lower than the service rate, so the ASPs tend to deploy fewer IPS VMs. However, when the number of malicious users increases, the deployed IPS VMs can intercept more malicious requests. As the number of malicious users increases, the behavior of the ASP changes at a specific point, which is the point where the interception rate and the service rate intersects. Except this intersection point, the rest results are similar to the high-workload case.

### 6.4 Comparison Between Various IPS Efficiency

**ASP with high workload EUs:** Similar to the case Section 6.3, the ASPs change their best responses when the IPS efficiency increases. When ASPs have high workload EUs, by Equation (5), the service rate would be low. Therefore, the interception rate would be higher than the service rate in default parameters ( $\nu = 50$ ), and the ASPs tend to deploy more IPS VMs. However, when the IPS efficiency is low, the interception rate is lower, and the service rate is higher. As the efficiency of IPS increases, the best responses of the

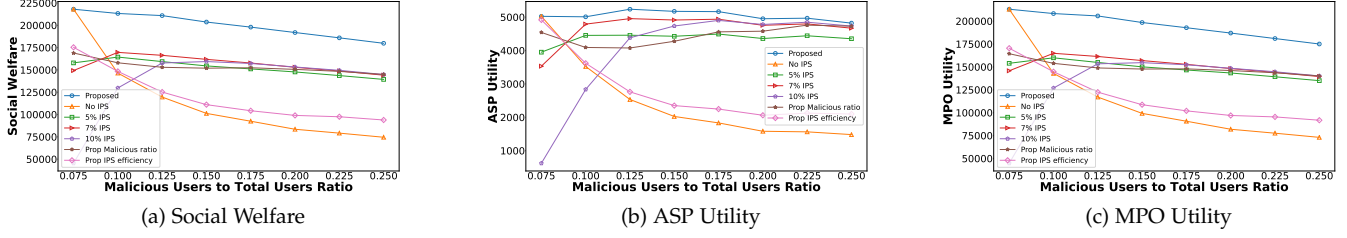


Fig. 7: Different Malicious Users to Total Users Ratio of High Workload EUs

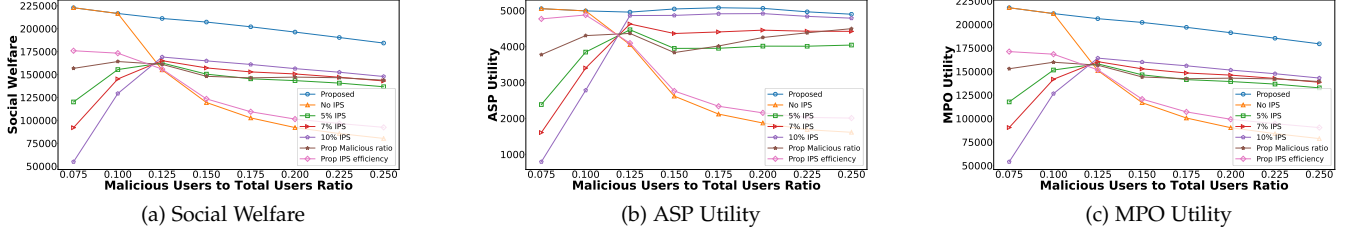


Fig. 8: Different Malicious Users to Total Users Ratio of Low Workload EUs

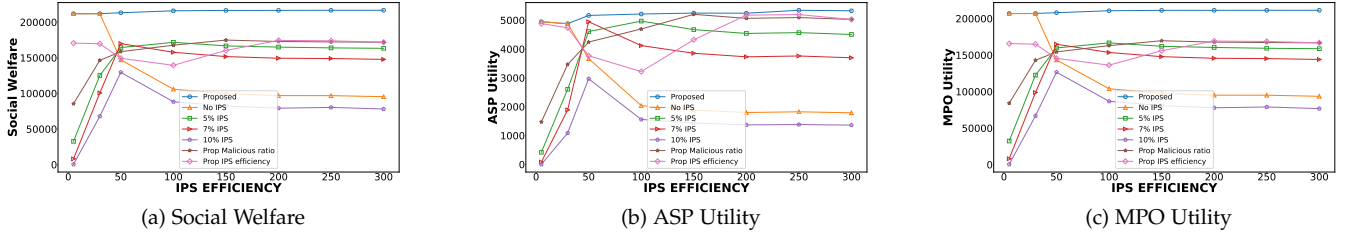


Fig. 9: Different IPS efficiency of High Workload EUs

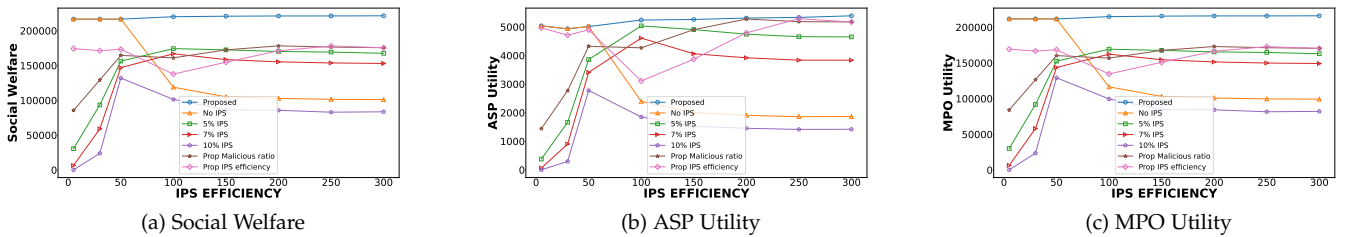


Fig. 10: Different IPS efficiency of Low Workload EUs

ASPs change at a specific point, which is the point where the interception rate and the service rate intersect.

When the IPS efficiency is low, the ASPs tend to deploy fewer IPS VMs. Thus, the scheme *No IPS* has higher utility than *5% IPS VM* and *7% IPS VM*. When the utility of fixed ratio schemes such as *5% IPS VM* and *7% IPS VM* increases as the IPS efficiency increases, the intercepted malicious request also increases. However, the utilities of these schemes started to decrease at some point. The main reason is that, as the IPS efficiency increases, ASPs need fewer IPS VMs to defend against these attacks, and these schemes deploy redundant IPS VMs. As for *10% IPS VM*,

the scheme always deploys redundant IPS VMs. Therefore, the utility is always lower than other schemes.

In Figure 9b, with regard to the scheme *Proportional IPS Efficiency*, its utility is higher than *5% IPS VM* and *7% IPS VM* when the IPS efficiency is low because it deploys fewer IPS VMs, which is  $0.01 \times 30\% = 0.3\%$ . As the IPS efficiency increases, the utility of these fixed ratio schemes *5% IPS VM* and *7% IPS VM* begins to surpass the utility of *Proportional IPS Efficiency* because these fixed schemes deploy more IPS VMs, which can intercept more malicious requests. As the IPS efficiency continues to rise, the utility of *Proportional IPS Efficiency* starts to exceed that of *5%*

IPS VM and 7% IPS VM because these fixed ratio schemes deploy redundant IPS VMs, while *Proportional IPS Efficiency* deploys the appropriate number of IPS VMs.

**ASP with low workload EUs:** When ASPs have low workload EUs, by Equation (5), the service rate would be high. Therefore, the interception rate would be lower than the service rate in default parameters ( $\nu = 50$ ), and the ASPs tend to deploy fewer IPS VMs. As the efficiency of IPS increases, the best responses of the ASPs change at a specific point, which is the point where the interception rate and the service rate intersect. The main difference with the high-workload case is that the interception and service rates intersect at a point where the IPS efficiency is higher.

## 7 DISCUSSION

More factors can be considered in the resource allocation problem for IPS and the main application service.

**Resource Reservation:** ASPs may require additional computational resources to operate and run necessary applications in the background, such as video caching services in streaming applications [51], which may occupy the resource for IPS. To incorporate this requirement into our system model, we introduce a new symbol  $\delta_i$ , which represents the amount of computational resources that must be reserved for each ASP  $i$ . In that case, we can redefine the process time Equation (6) as

$$T_i^p(z_i^v, z_i^h) = \frac{1}{(z_i^v - \delta_i - z_i^h)\mu_i^v - (\lambda_i - H_i(z_i^h))} \quad (54)$$

Although the outcome of the resource allocation may change when reserving additional resources, all of the observations and conclusions remain valid.

**Variable Rate for Malicious Attacks:** The rate of DDoS attacks is another important aspect to consider. In this study, we assumed that the rate of malicious traffic denoted as  $\lambda_{ij}, j \in U_i^m$  remains constant over time. However, it is possible that the rate of attacks may fluctuate over time [52]. We can redefine the rate of malicious attacks as a function of time, denoted as  $\lambda_{ij}(t), j \in U_i^m$ . Also, we can use  $\lambda_i^m(t) = \sum_{j \in U_i^m} \lambda_{ij}(t)$  to represent the time-dependent malicious attack rate on ASP  $i$ . In this new setting, we can divide the time into several time slots and then apply the proposed algorithm to each time slot. The time-dependent best response can then be derived accordingly.

## 8 CONCLUSION

In the paper, we proposed a flexible IPS deployment strategy for ASPs to allocate VM resources for IPS and services to mitigate the expected DDoS attacks. The joint optimization problem of ASP and MPO is formulated as a Stackelberg game to capture the hierarchy relationship. The optimal pricing strategy of MPO, the optimal number of purchased VMs, and the ratio of IPS VM for ASPs are derived. The Stackelberg equilibrium can then be derived through the proposed algorithm. The simulation results verify the proposed solution can maintain service quality and social welfare compared with other baseline schemes.

## REFERENCES

- [1] J. Singh, Y. Bello, A. R. Hussein, A. Erbad, and A. Mohamed, "Hierarchical security paradigm for iot multiaccess edge computing," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5794–5805, 2021.
- [2] S. T. Zargar, J. Joshi, and D. Tipper, "A survey of defense mechanisms against distributed denial of service (ddos) flooding attacks," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 2046–2069, 2013.
- [3] A. Praseed and P. S. Thilagam, "Ddos attacks at the application layer: Challenges and research perspectives for safeguarding web applications," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 661–685, 2019.
- [4] S. Ranjan, R. Swaminathan, M. Uysal, and E. Knightly, "Ddos-resilient scheduling to counter application layer attacks under imperfect detection," in *IEEE INFOCOM*, 2006, pp. 1–13.
- [5] N. Tripathi and N. Hubballi, "Application layer denial-of-service attacks and defense mechanisms: A survey," *ACM Comput. Surv.*, vol. 54, no. 4, may 2021. [Online]. Available: <https://doi.org/10.1145/3448291>
- [6] N. Agrawal and S. Tapaswi, "Defense mechanisms against ddos attacks in a cloud computing environment: State-of-the-art and research challenges," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3769–3795, 2019.
- [7] M. Nadeem, A. Arshad, S. Riaz, S. S. Band, and A. Mosavi, "Intercept the cloud network from brute force and ddos attacks via intrusion detection and prevention system," *IEEE Access*, vol. 9, pp. 152 300–152 309, 2021.
- [8] B. Siregar, R. F. Dwiputra Purba, Seniman, and F. Fahmi, "Intrusion prevention system against denial of service attacks using genetic algorithm," in *IEEE Comnetsat*, 2018, pp. 55–59.
- [9] Y. Chiang, Y. Zhang, H. Luo, T.-Y. Chen, G.-H. Chen, H.-T. Chen, Y.-J. Wang, H.-Y. Wei, and C.-T. Chou, "Management and orchestration of edge computing for iot: A comprehensive survey," *IEEE Internet of Things Journal*, vol. 10, no. 16, pp. 14 307–14 331, 2023.
- [10] Z.-L. Chang, C.-Y. Lee, C.-H. Lin, C.-Y. Wang, and H.-Y. Wei, "Game-theoretic intrusion prevention system deployment for mobile edge computing," in *IEEE GLOBECOM*, 2021, pp. 1–6.
- [11] M. E. Ahmed, S. Ullah, and H. Kim, "Statistical application fingerprinting for ddos attack mitigation," *IEEE Trans. on Information Forensics and Security*, vol. 14, no. 6, pp. 1471–1484, 2019.
- [12] A. Santoyo-González, C. Cervelló-Pastor, and D. P. Pazaros, "High-performance, platform-independent ddos detection for iot ecosystems," in *2019 IEEE 44th Conference on Local Computer Networks (LCN)*, 2019, pp. 69–75.
- [13] A. Ali and M. M. Yousaf, "Novel three-tier intrusion detection and prevention system in software defined network," *IEEE Access*, vol. 8, pp. 109 662–109 676, 2020.
- [14] Q. Yan, W. Huang, X. Luo, Q. Gong, and F. R. Yu, "A multi-level ddos mitigation framework for the industrial internet of things," *IEEE Communications Magazine*, vol. 56, no. 2, pp. 30–36, 2018.
- [15] Y. Shen, C. Wu, D. Kong, and M. Yang, "Tpdd: A two-phase ddos detection system in software-defined networking," in *IEEE ICC*, 2020.
- [16] K. Kalkan, L. Altay, G. Gür, and F. Alagöz, "Jess: Joint entropy-based ddos defense scheme in sdn," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 10, pp. 2358–2372, 2018.
- [17] L. D. Tsobdjou, S. Pierre, and A. Quintero, "An online entropy-based ddos flooding attack detection system with dynamic threshold," *IEEE Trans. on Network and Service Management*, vol. 19, no. 2, pp. 1679–1689, 2022.
- [18] J. Gao, L. Gan, F. Buschendorf, L. Zhang, H. Liu, P. Li, X. Dong, and T. Lu, "Omni scada intrusion detection using deep learning algorithms," *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 951–961, 2021.
- [19] Z. Liu, X. Yin, and Y. Hu, "Cpss lr-ddos detection and defense in edge computing utilizing dcnr q-learning," *IEEE Access*, vol. 8, pp. 42 120–42 130, 2020.
- [20] H. Zhang, J. Hao, and X. Li, "A method for deploying distributed denial of service attack defense strategies on edge servers using reinforcement learning," *IEEE Access*, vol. 8, pp. 78 482–78 491, 2020.
- [21] X. Tan, H. Li, L. Wang, and Z. Xu, "Global orchestration of cooperative defense against ddos attacks for mec," in *IEEE WCNC*, 2019, pp. 1–6.
- [22] Y. Deng, H. Jiang, P. Cai, T. Wu, P. Zhou, B. Li, H. Lu, J. Wu, X. Chen, and K. Wang, "Resource provisioning for mitigating



- edge ddos attacks in mec-enabled sdvn," *IEEE Internet of Things Journal*, pp. 1–1, 2022.
- [23] Y. Jia, F. Zhong, A. Alrawais, B. Gong, and X. Cheng, "Flowguard: An intelligent edge defense mechanism against iot ddos attacks," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9552–9562, 2020.
- [24] W.-C. Chang, Y. Chiang, Y. Zhang, and H.-Y. Wei, "Edge computing dynamic resource management: Tradeoffs between security and application qoe," in *IEEE VTC2021-Fall*, 2021, pp. 1–5.
- [25] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Trans. on Networking*, vol. 24, no. 5, pp. 2795–2808, 2016.
- [26] J. Myung, K. Kim, and Y.-J. Ko, "Transmit power control game with a sum power constraint in cell-free massive mimo down-link," *IEEE Wireless Communications Letters*, vol. 11, no. 3, pp. 632–635, 2022.
- [27] Y. Li, Z. Zhang, H. Wang, and Q. Yang, "Sers: Social-aware energy-efficient relay selection in d2d communications," *IEEE Trans. on Vehicular Technology*, vol. 67, no. 6, pp. 5331–5345, 2018.
- [28] K. Wang, M. Du, D. Yang, C. Zhu, J. Shen, and Y. Zhang, "Game-theory-based active defense for intrusion detection in cyber-physical embedded systems," *ACM Trans. Embed. Comput. Syst.*, vol. 16, no. 1, oct 2016. [Online]. Available: <https://doi.org/10.1145/2886100>
- [29] S. Ma, S. Guo, K. Wang, W. Jia, and M. Guo, "A cyclic game for service-oriented resource allocation in edge computing," *IEEE Trans. on Services Computing*, vol. 13, no. 4, pp. 723–734, 2020.
- [30] K. Wang, Z. Ding, D. K. C. So, and G. K. Karagiannidis, "Stackelberg game of energy consumption and latency in mec systems with noma," *IEEE Trans. on Communications*, vol. 69, no. 4, pp. 2191–2206, 2021.
- [31] Z. Liu, J. Su, Y.-a. Xie, K. Ma, Y. Yang, and X. Guan, "Resource allocation in d2d enabled vehicular communications: A robust stackelberg game approach based on price-penalty mechanism," *IEEE Trans. on Vehicular Technology*, vol. 70, no. 8, pp. 8186–8200, 2021.
- [32] Q. He, C. Wang, G. Cui, B. Li, R. Zhou, Q. Zhou, Y. Xiang, H. Jin, and Y. Yang, "A game-theoretical approach for mitigating edge ddos attack," *IEEE Trans. on Dependable and Secure Computing*, pp. 1–1, 2021.
- [33] J. Liu, X. Wang, S. Shen, G. Yue, S. Yu, and M. Li, "A bayesian q-learning game for dependable task offloading against ddos attacks in sensor edge cloud," *IEEE Internet of Things Journal*, vol. 8, no. 9, pp. 7546–7561, 2021.
- [34] H. Li, C. Yang, L. Wang, N. Ansari, D. Tang, X. Huang, Z. Xu, and D. Hu, "A cooperative defense framework against application-level ddos attacks on mobile edge computing services," *IEEE Trans. on Mobile Computing*, 2021.
- [35] L. Guo, Y. Xing, C. Jiang, and L. Bai, "A nfv-based resource orchestration algorithm for ddos mitigation in mec," in *IEEE IWCMC*, 2021, pp. 961–967.
- [36] Open Information Security Foundation, "Suricata." [Online]. Available: <https://github.com/OISF/suricata>
- [37] Y. Chi, T. Jiang, X. Li, and C. Gao, "Design and implementation of cloud platform intrusion prevention system based on sdn," in *IEEE ICBDA*, 2017, pp. 847–852.
- [38] D. Shi, L. Li, T. Ohtsuki, M. Pan, Z. Han, and H. V. Poor, "Make smart decisions faster: Deciding d2d resource allocation via stackelberg game guided multi-agent deep reinforcement learning," *IEEE Trans. on Mobile Computing*, vol. 21, no. 12, pp. 4426–4438, 2022.
- [39] N. Zhao, Y. Pei, Y.-C. Liang, and D. Niyato, "Multi-agent deep reinforcement learning based incentive mechanism for multi-task federated edge learning," *IEEE Trans. on Vehicular Technology*, vol. 72, no. 10, pp. 13 530–13 535, 2023.
- [40] T. Başar and G. J. Olsder, *Dynamic noncooperative game theory*. SIAM, 1998.
- [41] W. Zhang, X. Li, L. Zhao, X. Yang, T. Liu, and W. Yang, "Service pricing and selection for iot applications offloading in the multi-mobile edge computing systems," *IEEE Access*, vol. 8, pp. 153 862–153 871, 2020.
- [42] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA); User Equipment (UE) radio transmission and reception," 3GPP, Technical Specification (TS) 36.101, 4 2020, version 16.5.0.
- [43] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Trans. on Networking*, vol. 24, no. 5, pp. 2795–2808, 2015.
- [44] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA); Further advancements for E-UTRA physical layer aspects," 3GPP, Technical Report (TR) 36.814, 3 2017, version 9.2.0.
- [45] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3590–3605, 2016.
- [46] D. Ko, S. H. Chae, and W. Choi, "Mds coded task offloading in stochastic wireless edge computing networks," *IEEE Trans. on Wireless Communications*, vol. 21, no. 3, pp. 2107–2121, 2022.
- [47] W. Wu, P. Yang, W. Zhang, C. Zhou, and X. Shen, "Accuracy-guaranteed collaborative dnn inference in industrial iot via deep reinforcement learning," *IEEE Trans. on Industrial Informatics*, vol. 17, no. 7, pp. 4988–4998, 2021.
- [48] S. Diamond and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization," *Journal of Machine Learning Research*, vol. 17, no. 83, pp. 1–5, 2016.
- [49] A. Agrawal, R. Verschueren, S. Diamond, and S. Boyd, "A rewriting system for convex optimization problems," *Journal of Control and Decision*, vol. 5, no. 1, pp. 42–60, 2018.
- [50] A. Domahidi, E. Chu, and S. Boyd, "ECOS: An SOCP solver for embedded systems," in *European Control Conference (ECC)*, 2013, pp. 3071–3076.
- [51] S.-R. Yang, Y.-J. Tseng, C.-C. Huang, and W.-C. Lin, "Multi-access edge computing enhanced video streaming: Proof-of-concept implementation and prediction/qoe models," *IEEE Trans. on Vehicular Technology*, vol. 68, no. 2, pp. 1888–1902, 2019.
- [52] J. Mirkovic and P. Reiher, "A taxonomy of ddos attack and ddos defense mechanisms," *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 2, p. 39–53, apr 2004.



**Chun-Yen Lee** received the B.S. degree in electrical engineering and a minor degree in economics from the National Taiwan University (NTU) in 2021. He received the M.S. degree in computer science from the University of California, Los Angeles (UCLA). He was a research assistant with the Research Center for Information Technology Innovation, Academia Sinica. His research interests include wireless communications, game theory, and network economics.



**Chia-Hung Lin** received the B.S. degree in electrical engineering from National Taiwan University (NTU) in 2021. He was a research assistant of the Research Center for Information Technology Innovation, Academia Sinica. He received his M.S. degree in electrical and computer engineering at the University of Michigan, Ann Arbor. His research interests include wireless communications and embedded systems.



**Zhan-Lun Chang** received a B.S. degree in mathematics and electrical engineering and an MS in electrical engineering from the National Taiwan University (NTU), Taipei, Taiwan, in 2018 and 2020. He is currently pursuing Ph.D. degree in the Elmore Family School of Electrical and Computer Engineering (ECE) at Purdue University starting in 2021. His research interests include machine learning, artificial intelligence, and the analysis of next-generation networks, especially in exploring the integration of artificial

intelligence/machine learning with network optimization and resource allocation strategies.



**Chih-Yu Wang** is a Research Fellow / Professor in Research Center for Information Technology Innovation, Academia Sinica, Taiwan. He received the B.S. and Ph.D. degrees in electrical engineering and communication engineering from National Taiwan University (NTU), Taipei, Taiwan, in 2007 and 2013, respectively. He has been a visiting student in University of Maryland, College Park in 2011. His research interests include game theory, wireless communications, social networks, and data science.

He was a recipient of the Ta-You Wu Memorial Award from National Science and Technology Council, Taiwan in 2022, the Young Scholars' Creativity Award from Foundation for the Advancement of Outstanding Scholarship in 2021, and the Exploration Research Award 2021 from the Pan Wen-Yuan Foundation in 2021. He is an Associate Editor for IEEE Internet of Things journal. He is a IEEE Senior Member.



**Hung-Yu Wei** is a Professor in Department of Electrical Engineering and Graduate Institute of Communications Engineering, National Taiwan University. He served as Interim Department Chair and Associate Chair in Department of Electrical Engineering. He received the B.S. degree in electrical engineering from National Taiwan University in 1999. He received the M.S. and the Ph.D. degree in electrical engineering from Columbia University in 2001 and 2005 respectively. He was a summer intern at Telcordia

Applied Research in 2000 and 2001. He was with NEC Labs America from 2003 to 2005. His research interests include next-generation wireless networks, IoT, and fog/edge computing.

He has been actively participating in NGMN, IEEE 802.16, 3GPP, IEEE 1934, and IEEE 1935 standardization. He serves as Secretary for IEEE ComSoC Fog/Edge Industry Community. He serves as Chair of IEEE 1935 working group for edge/fog management and orchestration standard and Chair of IEEE 1934 Working Group for edge/fog computing and networking architecture. He is an Associate Editor for IEEE IoT Journal and IEEE IoT magazine. He was the Chair of IEEE VTS Taipei Chapter during 2016 2017.



## APPENDIX A

### PROOF OF LEMMA 1

*Proof.* The feasible region imposed by (11b) and (11c) is a closed and bounded interval in  $\mathbb{R}$  and thus is a convex set. As the objective function (11a) is continuous in  $z_i^h$ , the existence of the optimal solution then comes from the Extreme Value Theorem. If (11a) is concave at each feasible  $z_i^h$ , optimization problem (11) is a convex optimization problem. It remains to show that the objective function (11a) is concave in feasible  $z_i^h$ . By rearranging the terms, we express  $X_i(z_i^h|z_i^v, \Psi_m^v)$  as

$$\sum_{j \in \mathcal{U}_i^n} \Psi_{ij} \left( 1 - \frac{T_{ij}^t + T_i^p(z_i^v, z_i^h) - a_i}{b_i - a_i} \right) - \Psi_m^v z_i^v \quad (55)$$

To show  $X_i(z_i^h|z_i^v, \Psi_m^v)$  is concave in feasible  $z_i^h$ , it suffices to show that  $T_i^p(z_i^v, z_i^h)$  is convex in feasible  $z_i^h$ . We will verify the convexity of  $T_i^p(z_i^v, z_i^h)$ . From (6), We can rewrite it as

$$T_i^p(z_i^v) = \frac{1}{(z_i^v - z_i^h)\mu_i^v - \lambda_i + H_i(z_i^h)} \quad (56)$$

By definition,  $H_i(z_i^h)$  is concave in feasible  $z_i^h$ . In addition,  $(z_i^v - z_i^h)\mu_i^v - \lambda_i$  is an affine function of  $z_i^h$ , so  $(z_i^v - z_i^h)\mu_i^v - \lambda_i + H_i(z_i^h)$  is concave in feasible  $z_i^h$ . Given a positive concave function  $f(x)$  in feasible  $x$ ,  $\frac{1}{f(x)}$  is convex in feasible  $x$ . Thus,  $T_i^p(z_i^v, z_i^h)$  is convex in feasible  $z_i^h$  and  $X_i(z_i^h|z_i^v, \Psi_m^v)$  is concave in feasible  $z_i^h$ .  $\square$

## APPENDIX B

### PROOF OF LEMMA 2

*Proof.* From (19), we substitute  $\eta_i z_i^h$  into  $H_i(z_i^h)$ , then we can obtain

$$[(z_i^v - z_i^h) - \lambda_i + \eta_i z_i^h]^{-2} \cdot (-\mu_i^v + \eta_i) \quad (57)$$

Obviously, regardless of the value of  $z_i^h$ , the (57) will never be zero. Therefore, the extreme point does not exist.  $\square$

## APPENDIX C

### PROOF OF LEMMA 3

*Proof.* If  $T_{i,2}^p(z_i^v)$  is convex in feasible  $z_i^v$ ,  $Y_i^2(z_i^v|\Psi_m^v)$  is concave in feasible  $z_i^v$ . We will verify the convexity of  $T_{i,2}^p(z_i^v)$ . We can write  $T_{i,2}^p(z_i^v)$  as follow

$$T_{i,2}^p(z_i^v) = \frac{1}{(z_i^v - \min(\xi_i z_i^v, \frac{\lambda_i^m}{\eta_i}))\mu_i^v - (\lambda_i - \eta_i \min(\xi_i z_i^v, \frac{\lambda_i^m}{\eta_i}))} \quad (58)$$

By rewriting (58),  $T_{i,2}^p(z_i^v)$  can be further expressed as

$$T_{i,2}^p(z_i^v) = \frac{1}{z_i^v \mu_i^v - \lambda_i + (\eta_i - \mu_i^v) \min(\xi_i z_i^v, \frac{\lambda_i^m}{\eta_i})} \quad (59)$$

Since the term  $(\eta_i - \mu_i^v)$  is always positive in Case 2,  $(\mu_i^v - \eta_i) \min(\xi_i z_i^v, \frac{\lambda_i^m}{\eta_i})$  is positive and concave in feasible  $z_i^v$ . In addition,  $(z_i^v \mu_i^v - \lambda_i)$  is an affine function in feasible  $z_i^v$ , so  $z_i^v \mu_i^v - \lambda_i + (\mu_i^v - \eta_i) \min(\xi_i z_i^v, \frac{\lambda_i^m}{\eta_i})$  is concave in feasible  $z_i^v$ .

Given a concave function  $f(x)$  in feasible  $x$ ,  $\frac{1}{f(x)}$  is convex in feasible  $x$ . Therefore,  $T_{i,2}^p(z_i^v)$  is convex in feasible  $z_i^v$  and  $Y_i^2(z_i^v|\Psi_m^v)$  is concave in feasible  $z_i^v$ .  $\square$

## APPENDIX D

### PROOF OF THEOREM 1

*Proof.* We begin by proving the feasible region is a convex set.  $\Upsilon_i$  is a convex set because  $\Upsilon_i$  is a ray starting from  $\frac{\lambda_i + \gamma_i}{\mu_i^v}$  to  $\infty$ . Since we have proved the convexity of  $Y_i^2(z_i^v|\Psi_m^v)$  in Lemma 3,  $\Upsilon_i^2$  is a convex set because any superlevel set of a concave function is a convex set. Since the intersection of two convex set is a convex set,  $\Upsilon_i \cap \Upsilon_i^2$  is a convex set. When  $\Upsilon_i \cap \Upsilon_i^2 = \emptyset$ , which means the ASP  $i$  cannot make the processing queue stable while has a positive utility no matter how many VMs ASP  $i$  buys from the MPO. In this situation, the best action of ASP  $i$  is not to buy any VM. By Lemma 3,  $Y_i^2(z_i^v|\Psi_m^v)$  is concave. When the extreme point of  $Y_i^2(z_i^v|\Psi_m^v)$  is smaller than  $\frac{\lambda_i + \gamma_i}{\mu_i^v}$ ,  $\frac{\lambda_i + \gamma_i}{\mu_i^v}$  is the optimal point. When the extreme point of  $Y_i^2(z_i^v|\Psi_m^v)$  falls in the interior of the feasible region, the extreme point is the optimal point. The extreme point of  $Y_i^2(z_i^v|\Psi_m^v)$  can be solved when the first derivative of  $Y_i^2(z_i^v|\Psi_m^v)$  with respect to  $z_i^v$  equals zero.

$$-\frac{\partial T_{i,2}^p(z_i^v)}{\partial z_i^v} = \frac{\Psi_m^v(b_i - a_i)}{\sum_{j \in \mathcal{U}_i^n} \Psi_{ij}} \quad (60)$$

Using (21a), (21b), (22), and  $H_i(x) = \eta_i x$ , it only takes simple algebraic operations to have the expression for the extreme point in (24). Note that since there are two kinds of  $Y_i^2(z_i^v|\Psi_m^v)$  under the circumstances of (21a) and (21b), there are also two different extreme points (25) and (26) based on the  $Y_i^2(z_i^v|\Psi_m^v)$ .  $\square$

## APPENDIX E

### PROOF OF LEMMA 4

*Proof.* If  $T_{i,3}^p(z_i^v)$  is convex in feasible  $z_i^v$ ,  $Y_i^3(z_i^v|\Psi_m^v)$  is concave in feasible  $z_i^v$ . We can verify the convexity of  $T_{i,3}^p(z_i^v)$  by taking second derivative of  $T_{i,3}^p(z_i^v)$  with respect to  $z_i^v$ .

$$[2(\mu_i^v)^2](z_i^v \mu_i^v - \lambda_i)^{-3} \quad (61)$$

The first term of (61) is positive. Moreover,  $z_i^v \mu_i^v - \lambda_i$  is larger than 0 to satisfy the constraint (11b), so the second term of (11b) is always positive. Therefore, (61) is non-negative, which means  $T_{i,3}^p(z_i^v)$  is convex in feasible  $z_i^v$ . Hence,  $Y_i^3(z_i^v|\Psi_m^v)$  is concave in feasible  $z_i^v$ .  $\square$

## APPENDIX F

### PROOF OF THEOREM 2

*Proof.* We begin by proving the feasible region is a convex set.  $\Upsilon_i$  is a convex set because  $\Upsilon_i$  is a ray starting from  $\frac{\lambda_i + \gamma_i}{\mu_i^v}$  to  $\infty$ . Since we have proved the convexity of  $Y_i^3(z_i^v|\Psi_m^v)$  in Lemma 4,  $\Upsilon_i^3$  is a convex set because any superlevel set of a concave function is a convex set. Since the intersection of two convex set is a convex set,  $\Upsilon_i \cap \Upsilon_i^3$  is a convex set. When  $\Upsilon_i \cap \Upsilon_i^3 = \emptyset$ , which means the ASP  $i$  cannot make the processing queue stable while has a positive utility no matter how many VMs ASP  $i$  buys from the MPO. In this situation, the best action of ASP  $i$  is not to buy any VM. By Lemma 4,  $Y_i^3(z_i^v|\Psi_m^v)$  is concave. When the extreme point of  $Y_i^3(z_i^v|\Psi_m^v)$  is smaller than  $\frac{\lambda_i + \gamma_i}{\mu_i^v}$ ,  $\frac{\lambda_i + \gamma_i}{\mu_i^v}$  is the

optimal point. When the extreme point of  $Y_i^3(z_i^v | \Psi_m^v)$  falls in the interior of the feasible region, the extreme point is the optimal point. The extreme point of  $Y_i^3(z_i^v | \Psi_m^v)$  can be solved when the first derivative of  $Y_i^3(z_i^v | \Psi_m^v)$  with respect to  $z_i^v$  equals zero.

$$-\frac{\partial T_{i,3}^p(z_i^v)}{\partial z_i^v} = \frac{\Psi_m^v(b_i - a_i)}{\sum_{j \in U_i^n} \Psi_{ij}} \quad (62)$$

Using (39), (40), and  $H_i(x) = \eta_i x$ , it only takes simple algebraic operations to have the expression for the extreme point in (42).  $\square$

## APPENDIX G PROOF OF THEOREM 3

*Proof.* We start by proving that  $\sum_{i \in \mathcal{I}} (z_i^v(\Psi_m^v))^*$  is convex in  $\Psi_m^v$  when  $\Psi_m^v \in \mathcal{M}^k, \forall k \in \mathcal{I}$ . In (52), the second, third and fourth term pertains to  $\Psi_m^v$ . As the summation of convex function is also a convex function, it suffices to prove that  $(z_{i,2}^{v,e''}(\Psi_m^v))^*$  is convex in  $\Psi_m^v$  for  $i \in \Omega_2^{k,e''}$ ,  $(z_{i,2}^{v,e'}(\Psi_m^v))^*$  is convex in  $\Psi_m^v$  for  $i \in \Omega_2^{k,e'}$  and  $(z_{i,3}^{v,e}(\Psi_m^v))^*$  is convex in  $\Psi_m^v$  for  $i \in \Omega_3^{k,e}$ . Taking the second derivative of (25) and (26) with respect to  $\Psi_m^v$ , we have

$$\frac{3}{4}(\Psi_m^v)^{-5/2} \sqrt{\frac{\sum_{j \in U_i^n} \Psi_{ij}}{\mu_i^v(b_i - a_i)}} \geq 0 \quad (63)$$

$$\frac{3}{4}(\Psi_m^v)^{-5/2} \sqrt{\frac{\sum_{j \in U_i^n} \Psi_{ij}}{(b_i - a_i)[(1 - \xi_i)\mu_i^v + \xi_i\eta_i]}} \geq 0 \quad (64)$$

In addition, take the second derivative of (42) with respect to  $\Psi_m^v$ , we have

$$\frac{3}{4}(\Psi_m^v)^{-5/2} \sqrt{\frac{\sum_{j \in U_i^n} \Psi_{ij}}{\mu_i^v(b_i - a_i)}} \geq 0 \quad (65)$$

As a result,  $(z_{i,2}^{v,e'}(\Psi_m^v))^*$ ,  $(z_{i,2}^{v,e''}(\Psi_m^v))^*$  and  $(z_{i,3}^{v,e}(\Psi_m^v))^*$  is convex in  $\Psi_m^v \in \mathcal{M}^k, \forall k \in \{1, \dots, J\}$ . Since the sub-level set of a convex function is a convex set, the feasible region resulting from (53b) is a convex set for any value  $Q_v$ . In addition, the feasible region of (53c) is also a convex set because the convex combination of two numbers in  $\mathcal{M}^k, \forall k \in \{1, \dots, J\}$  is still a number in  $\mathcal{M}^k, \forall k \in \{1, \dots, J\}$ . Thus, the feasible region of the optimization problem (53) is the intersection of two convex set and thus is a convex set. To prove that the MPO's the optimization problem (53) is a convex optimization problem, what remains to do is to prove that the objective function (53a) is concave in  $\Psi_m^v \in \mathcal{M}^k, \forall k \in \{1, \dots, J\}$ . Using (24), (42) and (52), we can unfold the first term in (53a) as follows.

$$\begin{aligned} & \Psi_m^v \cdot \sum_{i \in \mathcal{I}} (z_i^v(\Psi_m^v))^* = \\ & \Psi_m^v \cdot \sum_{i \in \Omega^{k,l}} \frac{\lambda_i + \gamma_i}{\mu_i^v} + \Psi_m^v \cdot \sum_{i \in \Omega_2^{k,e'}} (z_{i,2}^{v,e'}(\Psi_m^v))^* + \\ & \Psi_m^v \cdot \sum_{i \in \Omega_2^{k,e''}} (z_{i,2}^{v,e''}(\Psi_m^v))^* + \Psi_m^v \cdot \sum_{i \in \Omega_3^{k,e}} (z_{i,3}^{v,e}(\Psi_m^v))^* \end{aligned} \quad (66)$$

The first term in (66) is a linear function of  $\Psi_m^v$  which is both convex and concave. The second and third term (66) is concave in  $\Psi_m^v$ . To prove these properties, we check the concavity via the second-order test. The second derivative of the second term is

$$-\frac{1}{4}(\Psi_m^v)^{-3/2} \sum_{i \in \Omega_2^{k,e'}} \sqrt{\frac{\sum_{j \in U_i^n} \Psi_{ij}}{(b_i - a_i)[(1 - \xi_i)\mu_i^v + \xi_i\eta_i]}} \leq 0 \quad (67)$$

The second derivative of the third term is

$$-\frac{1}{4}(\Psi_m^v)^{-3/2} \sum_{i \in \Omega_2^{k,e''}} \sqrt{\frac{\sum_{j \in U_i^n} \Psi_{ij}}{\mu_i^v(b_i - a_i)}} \leq 0 \quad (68)$$

And the second derivative of the forth term is

$$-\frac{1}{4}(\Psi_m^v)^{-3/2} \sum_{i \in \Omega_3^{k,e}} \sqrt{\frac{\sum_{j \in U_i^n} \Psi_{ij}}{\mu_i^v(b_i - a_i)}} \leq 0 \quad (69)$$

Therefore, (66) is concave in  $\Psi_m^v$ . The second term of (53a),  $C_m^v(\sum_{i \in \mathcal{I}} (z_i^v(\Psi_m^v))^*)$ , is convex in  $\Psi_m^v$  as it is a composition of  $C_m^v$  and  $\sum_{i \in \mathcal{I}} (z_i^v(\Psi_m^v))^*$  where  $C_m^v$  is convex and non-decreasing, and  $\sum_{i \in \mathcal{I}} (z_i^v(\Psi_m^v))^*$  is convex in  $\Psi_m^v \in \mathcal{M}^k, \forall k \in \{1, \dots, J\}$ . Since a concave function minus a convex function yields a concave function, (53a) is concave in  $\Psi_m^v \in \mathcal{M}^k, \forall k \in \{1, \dots, J\}$ . In sum, the (53) in each interval  $\mathcal{M}^k, \forall k \in \{1, \dots, J\}$  is a convex optimization problem.  $\square$

## APPENDIX H PROOF OF THEOREM 4

*Proof.* In Case 2, we have analytical solutions for ASPs' optimal number of VMs to buy from the MPO by Theorem 1. The best response sets for ASPs defined in (16) are non-empty. Also, in Case 3, we have analytical solutions for ASPs' optimal number of VMs to buy from the MPO by Theorem 2. The best response sets for ASPs defined in (16) are non-empty.

With the optimal responses from the ASPs, the MPO's optimal price per VM  $\Psi_m^v$  exists because MPO's optimization problem is convex proved in Theorem 3 with a compact region; thus, the MPO's best response set defined in (18) is non-empty. By Definition 1, there is at least one Stackelberg game equilibrium.  $\square$