An Auction-Based Mechanism for Task Offloading in a Secure Fog-Cloud Environment

Branka Mikavica(✉) and Aleksandra Kostić-Ljubisavljević

Faculty of Transport and Traffic Engineering, University of Belgrade, Belgrade, Serbia
b.mikavica@sf.bg.ac.rs

Abstract. With the rapid growth of terminal devices, fog-cloud systems become promising for solving delays and congestions in task provisioning. When the fog computing resources are insufficient, task offloading to the remote cloud can be performed to improve performances. Auctions are considered as a convenient tool to provide incentives for task offloading, and efficiently allocate resources in the fog-cloud environment. However, due to virtualization, the cloud segment in the fog-cloud system is prone to malicious attacks. Security requirements are often in contrast with performance requirements since the operation of security mechanisms consumes a part of computation capacities. Therefore, a comprehensive study is needed to address resource allocation with a security assessment. In this paper, we propose a novel simulation model for resource allocation in the hierarchical fog-cloud system with task offloading. To improve the experience and prevent performance deterioration, we introduce task differentiation into delay-sensitive and delay-tolerant tasks. Resource allocation of the fog and the cloud layer is based on the truthful double auction. A Vicrey-Clarke-Grove (VCG) driven resource allocation is established for winner determination. The proposed simulation model is used to analyze utility functions in the observed fog-cloud environment depending on the offered security level.

Keywords: Fog-cloud environment · Auction-based offloading · Security

1 Introduction

The number of Internet of Things (IoT) entities and IoT related services shows ever-increasing expansion [1]. These devices generate enormous data that need to be processed. The majority of IoT entities have limited computation and storage capacities. Hence, submitted tasks are sent to remote cloud data centers for processing and analysis. However, numerous tasks running on the cloud have specific deadlines. Furthermore, the networks' bandwidth is limited, thus posing an additional challenge for delay-sensitive and context-aware IoT applications [2]. To overcome these issues, computation offloading can be applied. Tasks from IoT entities are first sent to fog nodes, instead of being submitted directly to the cloud data center. The fog node executes the submitted task if there are sufficient computing resources. Otherwise, the task will be offloaded to the remote cloud. Introduction of fog provides timely-response service with local awareness

D. Perakovic and L. Knapcikova (Eds.): FABULOUS 2022, LNICST 445, pp. 47–63, 2022.
https://doi.org/10.1007/978-3-031-15101-9_4

and supports mobility and wireless access. The offloading problem can be analyzed at three levels, including the IoT entities layer, the fog layer, and the cloud layer. The aim of offloading is to optimize operations so that total costs are reduced [3]. In general, offloading depends not only on the single entity decision but also on the interactions of other entities. It is considered that fog nodes are reluctant to offload tasks to cloud nodes, thus deteriorating network performances. To solve this issue, fog nodes should be motivated to participate in offloading operations.

Due to virtualization, the security of a cloud segment in the fog-cloud system is an important issue to be addressed. In general, cloud resources are organized as pre-configured Virtual Machines (VMs), individually accessible over the Internet. The majority of cloud-specific attacks occur via compromised VM. There are numerous types of VM-based attacks with different effects on the VM under attack [4]. Delays and failures in task provisioning are possible. Depending on the attacks' severity, the number of available VMs can be reduced. Implementation of a security mechanism may be a promising solution to alleviate the performances degradation. However, security mechanisms consume some computation resources and sometimes may extend the processing time. Therefore, an appropriate security evaluation is needed to support efficient resource allocation.

Auction-based mechanisms are often recommended as a promising solution to improve efficiency and support fair distribution in various edge-computing domains, including mobile-edge computing, content delivery networks, fog computing, and cloud computing [5–8]. In an auction process, there are three essential participants: buyer, seller, and auctioneer. The buyer aims at acquiring resources with minimum expenses. The seller offers its resources with the main goal to maximize revenues by selling resources to buyers that value them the most. The auctioneer acts as an intermediate agent that determines the winners. There are numerous types of auction mechanisms [9], including English auction, Dutch auction, Vickrey auction and its modification to Vicrey-Clarke-Grove (VCG) auction, combinatorial, double, etc. It is considered that double auctions can efficiently balance buyers' and sellers' benefits. During a double auction process, buyers place bids (buyers' willingness to pay), while sellers place asks (the required price). If the bids are greater than or equal to the ask, an auctioneer settles the hammer price, and the process terminates. Auctions are also proposed as an efficient tool to provide incentives for offloading in a fog-cloud environment [3].

The main contributions of this paper are the following: (i) we formulate the system model that comprises an IoT layer, a fog layer, and a cloud layer; to allocate resources in both fog and cloud layer, double auction-based mechanisms are performed in the two stages: between IoT layer and fog layer (I stage), and between fog layer and cloud layer (II stage); (ii) we set truthful VCG-based double auctions that satisfy individual rationality and provide incentives for task offloading; (iii) to improve experience and resource utilization in task provisioning on the fog layer, we distinguish delay-sensitive and delay-tolerant tasks; the highest priority in task provisioning on fog layer is assigned to delay-sensitive tasks, thus preventing performance degradation; (iv) a VM security modeling on the cloud layer in introduces, with the malicious VM-based attack and the complexity of the security mechanism as critical factors affecting the VMs availability.

The remainder of the paper is organized as follows. Section 2 reviews related works on the problems of auction-based offloading optimization and resource allocation in a fog-cloud environment. The system model and problem formulation comprising cloud

layer security modeling, establishing double VCG-based auction-mechanisms between the layers, and the utility functions formulations are presented in Sect. 3. In Sect. 4, we present performance evaluation and discuss simulation results. Finally, Sect. 5 provides concluding remarks and future research directions.

2 Related Work

Auction-based resource allocation in a fog environment is addressed in various studies. In [10], a decentralized auction-based fog node allocation that improves the utilization of fog resources is proposed. The major advantage of the proposed model is that allows fog resource providers to participate in more than one auction simultaneously. Moreover, the model reduces the number of exchanged messages and represents an application of 5G due to decentralization and requirements in terms of bandwidth. In [11], the authors propose two types of truthful mechanisms for resource allocation and pricing in a fog environment, Fixed Price based Fog Node Allocation Mechanism, and Combinatorial Auction based Fog Service Allocation Mechanism. It shows that a combinatorial auction-based mechanism can improve resource allocation with high proficiency and higher revenues for fog providers. A dynamic resource allocation model based on the overbooking mechanism is proposed in [12]. The model considers the Quality of Service (QoS) requirements and provides individual rationality, computation efficiency, and truthfulness. The results show that the auction achieves the preferred properties, and the given resource allocation maximizes the profit of nodes with a high degree of QoS satisfaction.

Cloud resource allocation and pricing strategies using auction-based mechanisms are widely studied in the literature. In [13], a comprehensive survey on auction-based mechanisms in a cloud environment is provided. VCG auction mechanism is very often used since it provides a socially optimal solution [14]. Relations between security and cloud resource allocation are analyzed in [15, 16]. An auction-based mechanism that provides incentives for customers to reveal their actual requests and security valuations is proposed in [15]. The mechanism applies a greedy allocation rule, where customers are prioritized depending on their valuation of the security. The results show acceptable performance compared to the offline VCG-based auction mechanism. A truthful VCG-based auction mechanism addressing revenues, security, and energy consumption in a cloud environment is proposed in [4]. The VMs security model is proposed to assess the security level of VMs. The simulation results show that investment in security increases revenues and reduces rejection rate, but concurrently, increases energy consumption and the provider's lost revenue.

Numerous studies addressed the problem of offloading in fog computing. Comprehensive surveys on fog-cloud offloading are provided by [17, 18]. Auction mechanisms are often used to provide incentives for participants in the fog-cloud system to offload tasks. Thus, an incentive-compatible offloading mechanism in a fog-cloud environment can be performed by using a second-price sealed-bid auction [3]. The proposed system considers fog nodes and cloud data centers as bidders and auctioneers, respectively. The observed auction mechanism provides incentive compatibility and individual rationality. The problem is formulated using the queuing theory in both the edge layer and the cloud

layer. The results show that the method proposed in [3] outperforms other state-of-the-art methods in terms of execution time, energy consumption, and network usage.

An ascending-bid auction mechanism is proposed in [9] to relieve the strict requirements of cloud computing on delays, congestion and energy consumption. The fog network is divided into several clusters. In each cluster, a fog controller acts as an auctioneer and schedules the idle fog nodes for task provisioning. The proposed model guarantees the QoS requirements of task nodes. The fog controller takes the reward prices as their strategy, while the fog nodes take task sizes as their strategies. The utility function on the fog nodes is proposed, with the analysis of the cost of task computation delay and energy consumption. The results show satisfactory performance and a win-win solution under the condition of meeting the QoS.

An auction-based optimization method for modeling the offloading interactions in a fog-cloud environment is proposed in [19]. The interactions between fog nodes and the cloud entities are modelled with consideration of the specifications and limitations of the underlying physical infrastructure, with the bandwidth as a commodity.

A multi-attribute combinatorial reverse auction-based model for resource allocation in a system that includes customer, auctioneer, fog provider, cloud provider, and fog & cloud provider together as auction participants, is proposed in [20]. The model distinguishes three types of resources, local fog, remote fog, and cloud. The proposed auction model is a truthful, robust, and fair allocation method that considers response time, data source mobility requirements, and fog resource limitations. To support truthful bidding, the Vicrey model is extended.

In this paper, the proposed simulation model analyzes resource allocation, security, and task offloading in the fog-cloud environment. To the best of our knowledge, this is the first paper addressing the following issues jointly: (i) truthful double auctioning between all participants in the fog-cloud system; (ii) cloud layer security modeling, where the complexity of the security mechanism and malicious attacks are used to assess the security level in various scenarios; (iii) introducing prioritization in the offloading task process to the remote cloud.

3 System Model and Problem Formulation

As shown in Fig. 1, the hierarchical architecture of the proposed system comprises three layers: the IoT entities layer, the fog layer, and the cloud layer.

The IoT entities include mobile devices, sensors, or end devices that require computation resources to run their heavy computation tasks.

The fog layer contains fog nodes. In general, fog nodes can be categorized as Fog Computational Nodes (FCNs) and Fog Broker Nodes (FBNs). FCNs provide the computation resource to run tasks of IoT entities. Compared to IoT entities, these nodes have some moderate computation and storage capabilities. FBNs act as mediators between FCNs and IoT entities [10]. It is assumed that the fog layer provides the highest level of security.

The cloud layer comprises cloud resources and Cloud Broker Nodes (CBNs). The resources are organized in the form of pre-configured virtual machines (VMs). VMs are individually accessible over the network. Hence, they are vulnerable to malicious

attacks. Attacks may cause delays and failure in tasks' provisioning, or reduce the number of available VMs, thus deteriorating performances. These effects may be alleviated by the implementation of security mechanisms. Depending on the security level provided, in this paper, we divide all VMs into several Trust Zones (TZs). The term TZ can be described as a combination of network segmentation and identity access management controls that define physical, logical, or virtual boundaries in network resources [21]. CBNs act as mediators between the fog layer and the cloud layer.

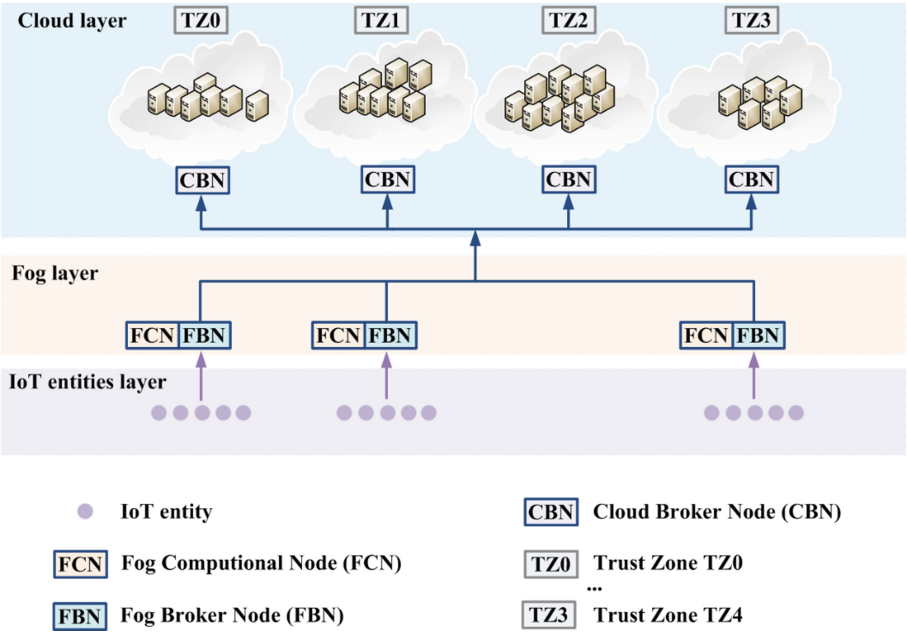

Fig. 1. The architecture of the secure fog-cloud system with task offloading

Resources on the fog and cloud layer are considered as a commodity for a double auction. IoT entities take the role of the bidders for task provisioning to parenting FCN. Each task is characterized depending on requirements in terms of delay and security level. Concurrently, FCNs place asks for fog resources via FBNs. Dedicated FBNs are auctioneers in the auction process. The role of FBNs is to match values of bids and asks, to prioritize the arriving tasks and proceed winning delay-sensitive tasks to FCNs for further processing. If the capacity of the FCN is not exceeded after allocation of the winning delay-sensitive tasks, the FBN can perform a double auction for delay-tolerant tasks and allocate those tasks for provisioning in the fog. The remaining delay-tolerant tasks are offloaded to the remote cloud. It should be emphasized that the fog layer is neutral in offloading since the offloading costs are considered while delegating tasks for offloading. Depending on the required security level, tasks are offloaded to the corresponding TZ. CBNs are auctioneers in auction-based task offloading.

3.1 Security Modeling in the Cloud Layer

Similar to [22], the term intensity will be used to describe the algorithm complexity or security level of the implemented security mechanism. TZs facilitate the management and tasks' processing in the cloud layer. An example of cloud network segmentation in TZs is shown in [4]. Similar security interpretation applies in Amazon Web Virtual Private Clouds (Amazon VPC) [23]. The security of a TZ is built upon an adequate configuration of domain controllers, firewalls, switches and routers, thus supporting segmentation and protecting access to cloud resources. Within a TZ, all VMs apply the unitive security mechanism with the same intensity. Thus, the TZ depicts a certain security level.

In this paper, we assume that all VMs are of the same computation and storage capabilities. The analysis is performed in N consecutive time slots. There are three assumed intensities for the corresponding security mechanism: high, middle and low. Higher intensity of the security mechanism implies higher complexity of the implemented algorithm, and consequently, higher security level. Four TZs can be distinguished depending on the provided security level, namely, TZ0, TZ1, TZ2 and TZ3. TZ0 provides the highest security level and all VMs in this TZ apply the security mechanism with high intensity. Accordingly, VMs in TZ1 and TZ2 use the security mechanism with medium and low intensity, respectively. TZ3 does not provide guarantees in terms of security, since VMs in this TZ do not apply a security mechanism.

It is considered that most attacks are unorganized, spontaneous, with a random arrival rate [22]. In this paper, the probability that a VM is threatened by a malicious attack is denoted by $p_a \in (0, 1)$. The probability that a VM remains available under an attack is denoted by $\beta_j \in (0, 1)$, where $j \in \{0, 1, 2, 3\}$ denotes the corresponding TZ. There are m_j available VMs in TZ j at the beginning of each time slot. It applies, $\beta_0 > \beta_1 > \beta_2 > \beta_3$, i.e., lower security level causes higher probability of a successful attack on a VM.

3.2 Auction Mechanisms

The dynamic resource allocation and pricing in the analyzed fog-cloud environment is performed using a two-stage VCG-based double auction mechanism. Thus, the double auction mechanisms are established between the IoT entities layer and the fog layer, and between the fog layer and the cloud layer.

VCG-Based Double Auction – I Stage. We assume there are F fog nodes with the role of the FCN. A FBN is assigned to each FCN in this architecture. The average number of IoT entities connecting to its parenting FCN is denoted by E. The number of active IoT entities generating request in time slot $i \in [1, N]$ for task execution on the FCN $f \in [1, F]$ is denoted by $E_{i,f}$. Furthermore, we assume that the arrival rate of requests for task execution follows a Poisson distribution with the average input rate λ [3].

All requests generated by IoT entities are computationally independent. To initiate task execution, an IoT entity creates and submits a bid to the FBN. We assume that each active IoT entity in a given time slot generates a single request for task execution, and each task can be executed within a single time slot. Each request that an active IoT entity generates in a given time slot i, $e_{i,f} \in \left[1, E_{i,f}\right]$ can be described as the following tuple:

$$e_{i,f} = \left(\tau_{e_{i,f}}, \psi_{e_{i,f}}, \theta_{e_{i,f}}, \upsilon_{e_{i,f}}\right) \tag{1}$$

The parameter $\tau_{e_{i,f}}$ in (1) is introduced to classify the task in terms of sensitivity to delays. Thus, $\tau_{e_{i,f}} = 0$ if the task can be classified as delay-sensitive. These tasks have the priority in the execution queue, so FCN primarily allocates its resources for their execution. If the task is delay-tolerant, it applies $\tau_{e_{i,f}} = 1$. Delay-tolerant tasks are executed on FCN if there are enough computing resources on the FCN. Otherwise, these tasks are offloaded to the cloud. The probability that task is delay-sensitive is denoted by ρ_s, while the probability that task is delay-tolerant is denoted by $\rho_t = 1 - \rho_s$.

To indicate preferred security level for task execution, the parameter $\psi_{e_{i,f}}$ is used in (1). Since the fog layer provides the highest security level, as indicated previously, it applies $\psi_{e_{i,f}} = 0$, if the task is delay-sensitive, i.e., $\tau_{e_{i,f}} = 0$, or the task is delay-tolerant, i.e., $\tau_{e_{i,f}} = 1$, but intended for execution on FCN f. If the task is delay-tolerant, and it is offloaded to the cloud, $\psi_{e_{i,f}}$ corresponds to the preferred TZ. Thus, for offloaded tasks it applies $\psi_{e_{i,f}} = j \in \{0, 1, 2, 3\}$. The probability that task will choose a certain security level for task provisioning is denoted by π_j, and it applies $\sum_j \pi_j = 1$.

The parameter $\theta_{e_{i,f}}$ in (1) is used to denote the resources that task $e_{i,f}$ occupies. If the task executes on FCN f, $\theta_{e_{i,f}}$ is expressed in capacity units. Here, the term capacity unit refers to the predefined unit of the computation or storage capacity. Moreover, we assume that each offloaded task requires a single VM for execution. Therefore, in the case of offloading, $\theta_{e_{i,f}} = 1$ VM.

The bid value is a nonnegative value, expressed by the parameter $\upsilon_{e_{i,f}}$ in (1), and represents the true willingness to pay per task for its execution. It should be noted that an IoT entity is not being charged by the bid value, but the value that is less or equal to the bid. The value to be paid is defined in an auction process. Bids are independent, and there is no information on others' bids. The assumed maximum bid value is the on-demand price for the given VM with appropriate security level, i.e., $\upsilon_{e_{i,f}} \in \left(0, p_{o,j}\right]$.

At the beginning of each time slot, each FBN defines FCN f asks per capacity unit for task provisioning in current time slot, denoted by $\alpha_{i,f} \in \left(\alpha_{i,f,\min}, \alpha_{i,f,\max}\right)$. Minimum ask represents the FCN f'cost per capacity unit for task provisioning. To prevent FCN f from posing to high asks per capacity unit, the maximum ask is limited to the $\alpha_{i,f,\max}$.

Once the bids are submitted by all active IoT entities, FBNs collect those bids and determine the set of winning bids for corresponding FCNs. At first, a FBN creates separate queues for delay-sensitive and delay-tolerant tasks for its dedicated FCN. Delay-sensitive tasks are of the highest priority and are intended for processing in the fog layer. To determine candidates for the set of winning delay-sensitive tasks, each FBN calculates the value of the unit bid for each task, i.e., the bid value per capacity unit of the FCN:

$$\nu_{e_{i,f}} = \frac{\upsilon_{e_{i,f}}}{\theta_{e_{i,f}}} \tag{2}$$

The candidates for the winning set are the tasks with unit bid is greater than or equal to the FCN f ask, i.e., the candidate is each task $e_{i,f}$ if it applies $\nu_{e_{i,f}} \geq \alpha_{i,f}$. Afterwards, the candidate tasks are sorted in the non-increasing order by the values of the unit bids. Tasks from the sorted set of candidate tasks are added to the winning set of tasks until there is enough capacity for task provisioning. The winning set of delay-sensitive tasks for the FCN f in the time slot i is denoted by $W_{i,f}^{S}$. Therefore, each winning delay-sensitive task

$\omega_{i,f}^S \in W_{i,f}^S$ satisfies the following:

$$\omega_{i,f,l}^S \geq \omega_{i,f,l+1}^S, \quad \nu_{\omega_{i,f,l}^S} \geq \alpha_{i,f}, \quad \sum \theta_{\omega_{i,f,l}^S} \leq Q \tag{3}$$

In (3), l denotes the position of the winning task in the $W_{i,f}^S$, and Q denotes the capacity of the FCN expressed in capacity units.

The price to be paid per task provisioning is determined using the VCG-based double auction. Each winning delay-sensitive task is charged depending on the unit bid value of the next winning task in the ordered set $W_{i,f}^S$ and the required resources. Thus, the price for each winning task in the $W_{i,f}^S$ can be expressed as:

$$p_{\omega_{i,f,l}^S} = \begin{cases} \nu_{\omega_{i,f,l+1}^S} \cdot \theta_{\omega_{i,f,l}^S}, & \textit{if } \exists \omega_{i,f,l+1}^S \\ \nu_{\omega_{i,f,l}^S} \cdot \theta_{\omega_{i,f,l}^S}, & \textit{otherwise} \end{cases} \tag{4}$$

As shown in (4), the price per task execution is always less than or equal to the value of the bid for the given task.

If the capacity of the FCN f is not exceeded by provisioning of delay-sensitive tasks, $\sum \theta_{\omega_{i,f,l}^S} \leq Q$, the dedicated FBN analyses the queue of the delay-tolerant tasks. To determine candidates for the set of winning delay-tolerant tasks that will be provisioned by FCN f, the FBN determines the unit bid per each delay-tolerant task using (2). Similar to the delay-sensitive tasks, the candidates for winning are the tasks with unit bid greater than or equal to the FCN f's ask per capacity unit. Afterwards, the candidate tasks are sorted in the nonincreasing order by the value of the unit bid. Tasks in this sorted list are added to the set of the winning delay-tolerant tasks by the value of the unit bid, from the highest to the lowest, until there are unutilized resources on FCN f available for task provisioning. The set of winning delay-tolerant tasks that are provisioned by FCN f is denoted by $W_{i,f}^T$. For each winning delay-tolerant task $\omega_{i,f}^T \in W_{i,f}^T$ applies:

$$\omega_{i,f,m}^T \geq \omega_{i,f,m+1}^T, \quad \nu_{\omega_{i,f,m}^T} \geq \alpha_{i,f}, \quad \sum \theta_{\omega_{i,f,m}^T} \leq Q, \tag{5}$$

where m in (5) denotes the position of the winning task in the $W_{i,f}^T$.

Similar to the case with the provisioning of delay-sensitive tasks by FCN f, the price to be paid per task provisioning is determined using the VCG-based auction. Each winning delay-sensitive task is charged depending on the unit bid value of the next winning task in the ordered set $W_{i,f}^T$ and the required resources. Thus, the price for each winning task in the $W_{i,f}^T$ can be expressed as:

$$p_{\omega_{i,f,m}^T} = \begin{cases} \nu_{\omega_{i,f,m+1}^T} \cdot \theta_{\omega_{i,f,m}^T}, & \textit{if } \exists \omega_{i,f,m+1}^T \\ \nu_{\omega_{i,f,m}^T} \cdot \theta_{\omega_{i,f,m}^T}, & \textit{otherwise} \end{cases} \tag{6}$$

If the FCN f's capacity Q is exceeded, the dedicated FBN collects all remaining delay-tolerant tasks, including those with unit bid lower than the FCN f ask, and creates a new queue intended for provisioning in the cloud layer.

VCG-Based Double Auction – II Stage. At the beginning of each time slot, dedicated CBNs determine asks per VM in each TZ j, denoted by $\gamma_{i,j} \in (\gamma_{i,j,\min}, \gamma_{i,j,\max})$. Minimum ask per task provisioning on a VM in the TZ j, $\gamma_{i,j,\min}$, represents the cost for task provisioning. To prevent too high asks, the upper bound for cloud asks is set to be $\gamma_{i,j,\max} = p_{o,j}$. Due to the same computation and storage capabilities, we assume that cloud provider sets the same asks for each VM in the TZ j.

Delay-tolerant task in a queue for offloading to the cloud can be expressed as follows:

$$e_{i,f}^{off} = \left(\psi_{e_{i,f}^{off}}, \upsilon_{e_{i,f}^{off}} \right) \tag{7}$$

In (7), $\psi_{e_{i,f}^{off}}$ denotes the requested security level for task provisioning, i.e., the selected TZ j, and $\upsilon_{e_{i,f}^{off}}$ denotes the bid value. These tasks are sorted in the nonincreasing order, not by the unit bid value, but their actual bid for the given task. The tasks intended for offloading to the cloud are tasks whose bids are greater than or equal to the FCN f's offloading cost per task, i.e., $\upsilon_{e_{i,f}^{off}} \geq \chi_{i,f}$. Thus, the fog layer does not incur losses due to task offloading. Each FBN delegates the tasks to the corresponding CBN, depending on the selected security level. Hence, each CBN generates the queue of the tasks intended for cloud processing on the selected TZ.

The set of candidate tasks comprises all tasks intended for offloading, whose bids are higher than the cloud provider's ask for given TZ, i.e., $\upsilon_{e_{i,f}^{off}} - \chi_{i,f} \geq \gamma_{i,j}$. The winning set for each TZ j, $W_{i,j}$, is defined depending on the length of the corresponding set of the candidate tasks, and the number of the available VMs.

The price to be paid per task provisioned on the cloud VM in the selected TZ is determined using the VCG-based double auction. Each winning task is charged by the value of the next highest bid (if the one exists), or the value of its bid, as long as there are available VMs in the given TZ. Therefore, the price for each winning task in the $W_{i,j}$ provisioned on the VM $k_j \in [1, m_j]$ can be expressed as follows:

$$p_{i,j,k_j} = \begin{cases} \omega_{i,j,k_{j+1}} \cdot \mu_{i,j,k_j}, & if \ k_j < |W_{i,j}| \\ \omega_{i,j,k_j} \cdot \mu_{i,j,k_j}, & if \ \ k_j = |W_{i,j}| \\ 0, & otherwise \end{cases} \tag{8}$$

In (8), we introduce the parameter μ_{i,j,k_j} to indicate VM availability, and it takes the value 0, if the VM k_j is unavailable due to the malicious attack in the current time slot; and the value 1, if the is no malicious attack, or the security mechanism prevented the failure due to the attack.

3.3 Utility Functions

We define utility functions for each layer in the analyzed hierarchical fog-cloud structure. In this paper, the term utility refers to the difference between the ask/bid for task provisioning and the revenue/cost. Due to the settings of the VCG-based double auctions established between the layers, the utility is always a non-negative value.

IoT Layer Utility. The utility function for the IoT layer comprises the utility for all provisioned delay-sensitive tasks on FCNs, the utility for all provisioned delay-tolerant tasks on FCNs, and the utility for tasks offloaded to the cloud:

$$U_{IoT} = U_{IoT}^{S} + U_{IoT}^{T} + U_{IoT}^{off} \tag{9}$$

The utility for provisioning of delay-sensitive tasks can be expressed as follows:

$$U_{IoT}^{S} = \sum_{i} \sum_{f} \sum_{l} \omega_{i,f,l}^{S} - p_{\omega_{i,f,l}^{S}} \tag{10}$$

The utility for provisioning delay-tolerant tasks by FCNs can be expressed as:

$$U_{IoT}^{T} = \sum_{i} \sum_{f} \sum_{m} \omega_{i,f,m}^{T} - p_{\omega_{i,f,m}^{T}} \tag{11}$$

The utility for tasks offloaded to the cloud can be expressed as follows:

$$U_{IoT}^{off} = \sum_{i} \sum_{j} \sum_{k_j} \gamma_{i,j} - p_{i,j,k_j} \tag{12}$$

Fog Layer Utility. The utility for the fog layer comprises utility for all delay-sensitive and delay-tolerant tasks provisioned by FCNs, as indicated by (13). It should be noted that the fog layer does not incur costs for tasks offloading to the cloud and obtains zero utility for offloading. Therefore, it is not included in the fog layer utility function.

$$U_{FOG} = U_{FOG}^{S} + U_{FOG}^{T} \tag{13}$$

The utility for provisioning delay-sensitive tasks by FCNs can be expressed as:

$$U_{FOG}^{S} = \sum_{i} \sum_{f} \sum_{l} \alpha_{i,f} - p_{\omega_{i,f,l}^{S}} \tag{14}$$

Accordingly, the utility for provisioning delay-tolerant tasks can be expressed as:

$$U_{FOG}^{T} = \sum_{i} \sum_{f} \sum_{m} \omega_{i,f,m}^{T} - p_{\omega_{i,f,m}^{T}} \tag{15}$$

Cloud Layer Utility. The utility for the cloud layer comprises utilities for tasks provisioned by VMs in each TZ, as indicated by (16). Compared to task provisioning on the fog layer, where the highest security level is provided, the cloud layer is exposed to potential malicious attacks, thus affecting the overall utility.

$$U_{Cloud} = \sum_{i} \sum_{j} \sum_{k_j} \gamma_{i,j} - p_{i,j,k_j} \tag{16}$$

4 Performance Evaluation

To analyze the proposed system model, we conducted a set of simulation experiments in the open-source programming language Python 3.7 in 100 iterations.

4.1 Simulation Setup

To investigate the utilities off all layers in the proposed VCG-based double auctions, and the effects of the cloud layer security, we set several scenarios. The analysis is performed in $N = 24$ time slots of one-hour duration. The number of fog nodes with the role of FCNs takes values from the set (25, 50, 75, 100). Since each FCN has a dedicated FBN, it applies the same number of FBNs. The average number of IoT entities connecting to its parenting FCN takes values from the set (10, 20, 30, 40, 50). There are 1200 available VMs available at the beginning of each time slot.

IoT Layer Setup. The number of active IoT entities initiating requests for tasks provisioning in each time slot is modeled by the Poisson distribution, with the average input rate $\lambda = 2$. The classification into delay-sensitive and delay-tolerant tasks is performed by the probability $\rho_s = \rho_t = 0.5$. The preferred security level is selected with the probability $\pi_j = 0.25$. The required FCN's resources per task are set randomly from the set (1 cu, 20 cu), where cu refers to the capacity unit. The bid value is also assigned randomly, with upper bound defined as the price for the equivalent on-demand VM instance.

Fog Layer Setup. For simplicity, we assume that all FCNs have the same computation and storage capabilities, with $Q = 100$ cu. Due to the same characteristics, the costs per capacity unit are the same for each FCN. Minimum ask for for task provisioning per capacity unit equals to the costs per capacity unit, and it applies $\alpha_{i,f,\min} = 0.05$\$/cu. To prevent seting too high asks in the double auction process, the upper bound for asks is set as $\alpha_{i,f,\max} = 0.25$ \$/cu. The cost of task offloading to the cloud is $\chi_{i,f} = 0.25$ per task.

Cloud Layer Setup. Depending on the VM allocation per TZ, we set 4 scenarios, as shown in Table 1.

Table 1. VMs allocation per TZ

Scenario	The number of VMs per TZ			
	TZ0	TZ1	TZ2	TZ3
The highest security	540	360	180	120
High security	420	280	140	360
Medium security	300	200	100	600
Low security	180	120	60	840

The probability of a VM malicious attack takes values from the set $p_a = \{0.1,\ 0.2,\ 0.3\}$. Simulation parameters relevant for TZs are listed in Table 2. Selected prices for equivalent on-demand VM instances are in the range of Amazon EC2 prices [24].

Table 2. Simulation parameters for TZs

Parameter	Trust Zones			
	TZ0	TZ1	TZ2	TZ3
Security mechanism's intensity	High	Medium	Low	–
Probability of VM's availability, β_j	0.84	0.75	0.66	0.5
VM's on-demand price per time slot [$/h]	32.576	16.288	8.144	4.072
VM's minimum ask per time slot [$/h]	9.773	4.886	2.443	1.222

4.2 IoT Entities Layer Utility

IoT entities layer utility mainly depends on the number of active IoT entities that generate requests for task provisioning. Figure 2. Shows the effects of variations in the number of connected IoT entities per FCN, for 50 FCNs. Overall utility for an IoT layer increases as the number of IoT entities per FCN increases. However, the utility significantly increases as the number of FCNs increases in the observed fog-cloud structure. Figure 3 shows the effects of variations of the number of FCNs in the given structure containing 30 connected IoT entities per FCN.

Since the large majority of requests are provisioned by FCNs, the utility slightly decreases for lower security provided. Thus, for the given number of IoT entities, the low-security scenario with the probability of 0.3 for malicious attack occurrence generates the lowest utility.

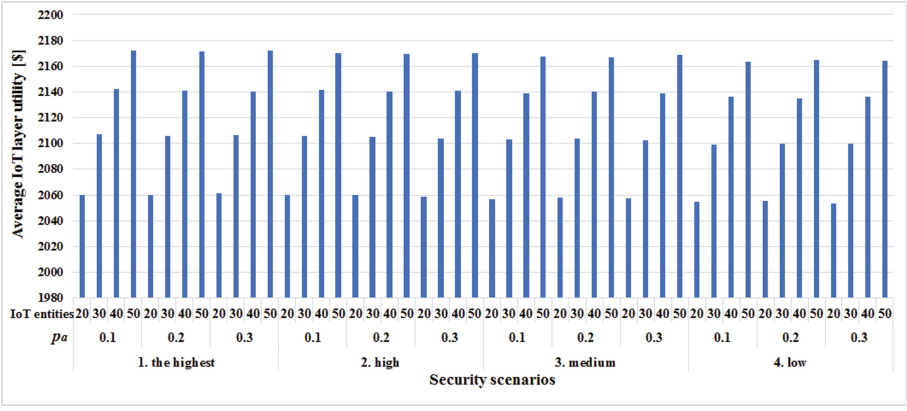

Fig. 2. Average utility of an IoT entities layer (50 FCNs)

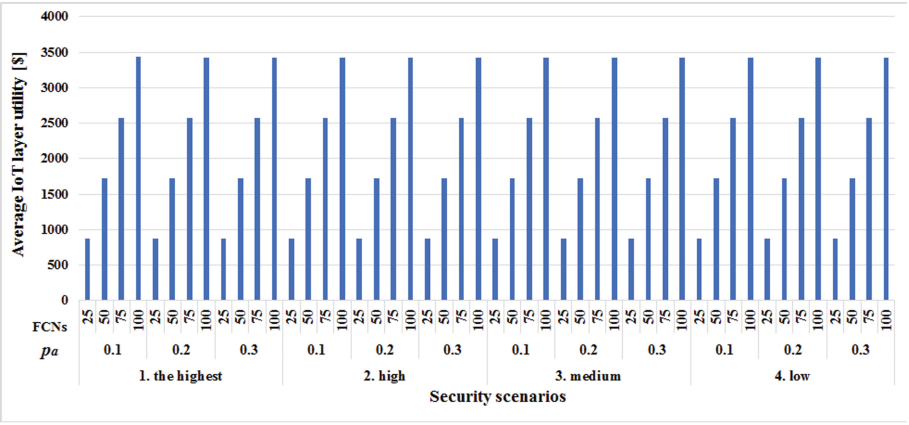

Fig. 3. Average utility of an IoT layer (30 IoT entities per FCN)

4.3 Fog Layer Utility

The fog layer, in general, is not affected by the security level provided by the cloud layer. However, the number of FCNs and the number of connected IoT entities induce utility to the large extent.

Figure 4 shows the average utility of a fog layer with variations in the number of FCNs in the fog-cloud system and variations in the number of connected IoT entities. Since overall security on the cloud layer does not affect the fog layer significantly, the results are shown for the medium-security scenario and the probability of malicious attack occurrence of 0.2. Notably, utility is directly proportional to the number of FCNs. As the number of connected IoT entities increases, the utility increases as well, but at a slower pace compared to the number of FCNs in the system.

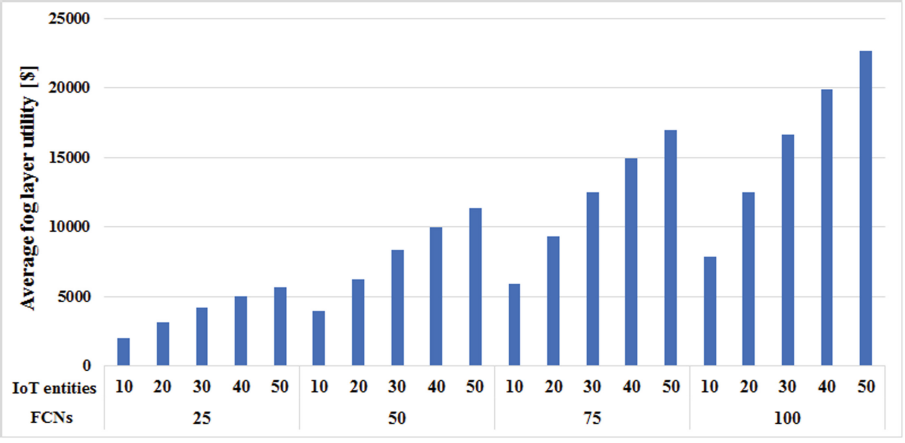

Fig. 4. Average utility of a fog layer

4.4 Cloud Layer Utility

Due to the VMs exposure and easy access over the Internet, the cloud layer is prone to malicious threats. The VMs vulnerability and effects on utility reduction are noticeable for both small and large network scenarios. As the probability of malicious attack occurrence increases, a low-security scenario is an undesirable solution.

Figure 5 shows the average cloud layer utility for 50 FCNs and variations in the number of connected IoT entities per FCN. A greater number of connected IoT entities per FCN significantly increases the cloud layer utility, especially for the highest and high-security scenarios.

Figure 6 shows the average cloud layer utility with 30 IoT entities per FCN and variations of the number of FCNs on the fog layer. The number of FCNs is the major driver of cloud layer utility enhancement. The utility is directly proportional to the number of FCNs in the network. Moreover, the investment in security is highly recommended, especially for large network scenarios.

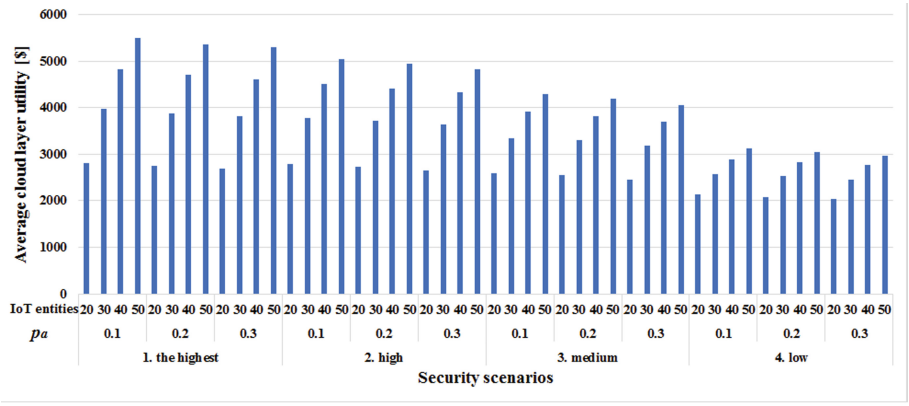

Fig. 5. Average utility of a cloud layer (50 FCNs)

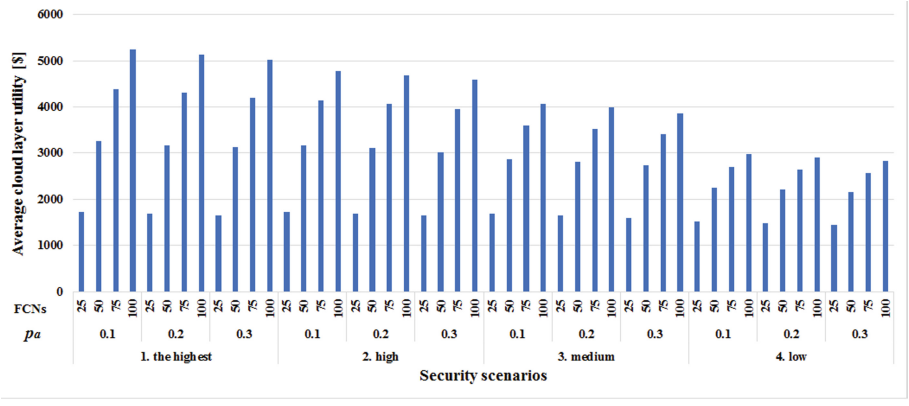

Fig. 6. Average utility of a cloud layer (30 IoT entities per FCN)

5 Conclusion

In this paper, the problem of task offloading in the fog-cloud environment is jointly addressed with security assessment and resource allocation. A truthful double VCG-based auction mechanism that provides individual rationality is established for winning tasks determination. The auction mechanism is performed in two stages. The first stage represents an auction between the IoT entities layer and the fog layer, while the second stage represents the auction process for task offloading to the cloud layer. The winning tasks in the first stage of the auction are executed in the fog. To improve performances, we introduce task prioritization on the fog layer depending on the delay requirements. Thus, winning delay-sensitive tasks are primarily executed. Offloading to the cloud requires security consideration. VM security modeling is introduced to assess the provided security level of VMs. The proposed simulation model is used to analyze the utility function of all participants in the fog-cloud environment.

Simulation results show that investment into security significantly increases the cloud layer utility in both small and large network scenarios. Due to task prioritization and forcing fog resources utilization improvements, the majority of the tasks are provisioned on the fog layer. Therefore, the utility of the IoT entities layer slightly decreases for lower security scenarios. The proposed model provides incentives for the fog layer to participate in task offloading since the operational offloading costs are covered in the auction process. Furthermore, the model intensifies resource utilization on the fog layer and offloads tasks to the remote cloud only in the fog resources are exceeded.

There are several future research directions. The proposed model can be extended to address potential penalties for failures. Also, the buffer size can be introduced in the fog and cloud layer to analyze effects on the execution time and network utilization. Another important issue to be solved is energy consumption. Since the operation of the security mechanism consumes computation resources, delays and energy consumption increase. Therefore, relations between energy consumption and network performance in the fog-cloud environment are subjects for future research.

References

1. Baranwal, G., Vidyarthi, D.: Admission control policies in fog computing using extensive form game. IEEE Trans. Cloud Comput., 1–14 (2020)
2. Singh, M., Baranwal, G.: Quality of service (QoS) in internet of things. In: 2018 3rd International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), pp. 1–6. IEEE. Bhimtal (2018)
3. Besharati, R., Rezvani, M.H., Sadeghi, M.M.G.: An incentive-compatible offloading mechanism in fog-cloud environments using second-price sealed-bid auction. J. Grid Comput. **19**(37), 1 (2021)
4. Mikavica, B., Kostic-Ljubisavljevic, A.: A security-driven approach for energy-aware cloud resource pricing and allocation. Adv. Electr. Comput. Eng. **21**(4), 99–106 (2021)

5. Wang, Q., Guo, S., Liu, J., Pan, C., Yang, L.: Profit maximization incentive mechanism for resource providers in mobile edge computing. IEEE Trans. Serv. Comput. **15**(1), 138–149 (2022)
6. Garmehi, M., Analoui, M., Pathan, M., Buyya, R.: An economic mechanism for request routing and resource allocation in hybrid CDN-P2P networks. Int. J. Netw. Manag. **25**(6), 375–393 (2015)
7. Tasiopoulos, A., Ascigil, O., Psaras, I., Toumpis, S., Pavlou, G.: FogSpot: spot pricing for application provisioning in edge/fog computing. IEEE Trans. Serv. Comput. **14**(6), 1781–1795 (2021)
8. Mikavica, B., Kostic-Ljubisavljevic, A.: Auction-based pricing in cloud environment. In: Khosrow-Pour, M. (eds.) Encyclopedia of Organizational Knowledge, Administration, and Technologies, pp. 86–97. IGI Global (2021)
9. Zu, Y., et al.: An auction-based mechanism for task offloading in fog networks. In: 2019 IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC), pp. 1–6. IEEE. Istanbul (2019)
10. Baranwal, G., Kumar, D.: DAFNA: decentralized auction based fog node allocation in 5G era. In: 2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS), pp. 575–580. IEEE. Rupnagar (2020)
11. Bandyopadhyay, A., Roy, T.S., Sarkar, V., Mallik, S.: Combinatorial auction-based fog service allocation mechanism for IoT applications. In: 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), pp. 518–524. IEEE. Noida (2020)
12. Zhang, F., Tang, Z., Chen, M., Zhou, X., Jia, W.: A dynamic resource overbooking mechanism in fog computing. In: 2018 IEEE 15th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), pp. 89–97. IEEE. Chengdu (2018)
13. Sheikholeslami, F., Navimipour, N.J.: Auction-based resource allocation mechanisms in the cloud environments: a review of the literature and reflection on future challenges. Concurrency and Comput. Pract. Experience **30**(16), 1–15 (2018)
14. Wang, X., Chen, X., Wu, W.: Towards truthful auction mechanisms for task assignment in mobile device clouds. In: IEEE Conference on Computer Communications (IEEE INFOCOM), pp. 1–9. IEEE. Atlanta (2017)
15. Halabi, T., Bellaiche, M., Abusitta, A.: Cloud security up for auction: a DSIC online mechanism for secure IaaS resource allocation. In: 2018 2nd Cyber Security in Networking Conference (CSNet), pp. 1–8. IEEE. Paris (2018)
16. Mikavica, B., Kostic-Ljubisavljevic, A., Popovic, D.: A security-driven approach to the auction-based cloud service pricing. Int. J. Transp. Traffic Eng. **11**(2), 213–228 (2021)
17. Yi, S., Li, C. and Li, Q.: A survey of fog computing: concepts, applications and issues. In: Proceedings of the 2015 Workshop on Mobile Big Data, pp. 37–42. ACM (2018)
18. Mahmud, R., Kotagiri, R., Buyya, R.: Fog computing: a taxonomy, survey and future directions. In: Di Martino, B., Li, K.-C., Yang, L.T., Esposito, A. (eds.) Internet of Everything. IT, pp. 103–130. Springer, Singapore (2018). https://doi.org/10.1007/978-981-10-5861-5_5
19. Besharati, R., Rezvani, M. H.: A prototype auction-based mechanism for computation offloading in fog-cloud environments. In: 2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI), pp. 542–547. IEEE. Tehran (2019)
20. Aggarwal, A., Kumar, N., Vidyarthy, D.P., Buyya, R.: Fog-integrated cloud architecture enabled multi-atributive combinatorial reverse auctioning framework. Simul. Model. Pract. Theory **109**(2021), 102307 (2021)
21. Gonzales, D., Kaplan, J., Saltzman, E., Winkelman, Z., Woods, D.: Cloud-trust – a security assessment model for infrastructure as a service (IaaS) clouds. IEEE Trans. Cloud Comput. **5**(3), 523–536 (2017)
22. Xu, H., Qiu, X., Sheng, Y., Luo, L., Xiang, Y.: A QoS-driven approach to the cloud service addressing attributes of security. IEEE Access **6**, 34477–34487 (2018)

23. Amazon Virtual Private Clouds (Amazon VPC). https://aws.amazon.com/vpc/?nc2=h_ql_prod_fs_vpc&vpc-blogs.sort-by=item.additionalFields.createdDate&vpc-blogs.sort-order=desc. Accessed 15 Dec 2021
24. Amazon EC2 On-Demand Pricing. https://aws.amazon.com/ec2/pricing/on-demand/. Accessed 15 Dec 2021