QoS-Aware Task Allocation and Scheduling in Three-Tier Cloud-Fog-IoT Architecture Using Double Auction

Nikita Joshi and Sanjay Srivastava
Dhirubhai Ambani Institute of Information and Communication Technology, Gandhinagar, India

Keywords: Internet of Things, Fog Computing, Task Scheduling, Perishable Resources, Double Auction, Competitive Bidding, Truthful Bidding.

Abstract: There is a lot of time-sensitive data provided by IoT applications that needs to be analysed. A Fog-Integrated Cloud Architecture is available to process these data. Due to the QoS requirements of IoT applications, the perishable nature of fog cloud resources, and the competition among users and service providers, task allocation in such an architecture is challenging. In this paper, we propose a competitive bidding (CompBid) strategy and a QoS-based task allocation and scheduling (QoTAS) algorithm using double auctions that aim to maximize user and service provider profit while also satisfying QoS requirements. A remote patient monitoring system is used to compare QoTAS performance to that of two previous studies, DPDA and MADA. In both the truthful and CompBid strategies, QoTAS achieves a higher task allocation ratio and resource utilization than DPDA and MADA. It has 89% more system utility than DPDA and 54% more user utility than MADA. Furthermore, the CompBid strategy increases QoTAS system utility by 25%.

1 INTRODUCTION

The benefits of various use cases of the Internet of Things (IoT) have resulted in a significant increment in IoT adoption. IoT applications generate large amount of time-critical data, that should be processed in a timely manner. Internet Database Connector (IDC) forecast estimated total of 41.6 billion connected IoT devices or "things" will generate 79.4 zettabytes (ZB) of data by 2025 (IoT, 2022). However, IoT devices cannot process all these data because of resource and energy constraints. Processing of this enormous volume of data typically takes place at remote cloud data centres (Suh et al., 2011).

Cloud servers have sufficient resources to process IoT data. However, the delay between cloud and IoT devices may not be acceptable by critical IoT applications such as healthcare and Vehicular Adhoc Network(VANET). Also, IoT applications generate large amount of data which requires hundreds of Gbps network bandwidth, which is more than traffic capacity of WAN in use. IoT applications need context-aware computing and communication in order to have intelligent interactions. Contextual data can include things like the user's location, actions, neighbours' gadgets, the time of day, etc. To deal with such IoT application requirements, fog computing concept is introduced in which network devices between IoT devices and cloud server are used for preprocessing, filtering and compressing the data.

In three tier Cloud-Fog-IoT architecture, IoT Service Providers (IoSPs) outsource the task to a Cloud Service Providers(CSPs) as shown in Figure 1. Fog Service Providers (FSPs) are registered with CSP. CSP allocates appropriate FSP to each IoSP. When allocating resources in such a distributed architecture, monetary cost must be considered. The cost of computing resources in edge computing, like cloud computing, is determined by their availability and usage (Jin et al., 2015).

According to J. Weinman, "fogonomics" is a branch of study that focuses on the economic aspects impacting the architecture of the fog computing system (Weinman, 2017). Fogonomics discusses a variety of pricing techniques, such as static price and dynamic price. In such a setting, dynamic pricing encourages healthy competition among FSPs, which is advantageous for both IoSPs and FSPs. However, dynamic pricing makes it challenging for FSPs to determine resource prices and for IoSPs to establish budgets dependent on the status of the market (Baranwal et al., 2018). The challenges of dynamic pricing are often overcome with the help of many modifications of the auction dynamic pricing method (Joshi and Sri-

Joshi, N. and Srivastava, S.
QoS-Aware Task Allocation and Scheduling in Three-Tier Cloud-Fog-IoT Architecture Using Double Auction.
DOI: 10.5220/0011967400003488
In *Proceedings of the 13th International Conference on Cloud Computing and Services Science (CLOSER 2023)*, pages 253-260
ISBN: 978-989-758-650-7; ISSN: 2184-5042

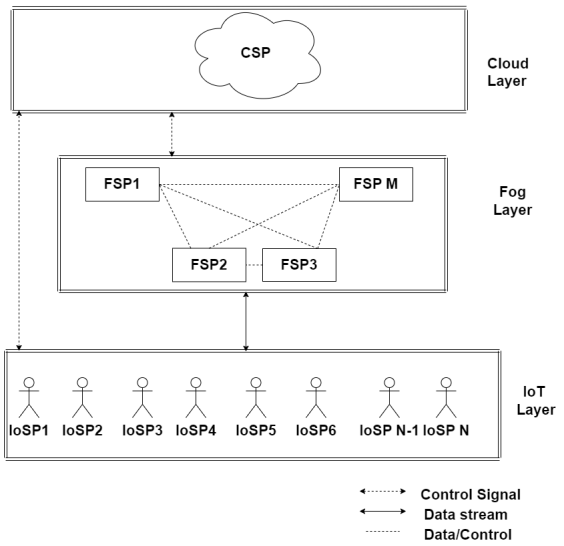


Figure 1: Fog IoT architecture.

vastava, 2019).

Money should not be the only consideration when allocating resources in this three-tiered design. Other factors to take into account include the QoS needs of IoT applications and the perishable nature of cloud and fog resources, which means they cannot be reused as double resources if not allocated at a time. It is possible that the FSP bids high in the in an effort to increase its profit, and no IoSP will be able to match that bid. It ultimately results in those resources not being auctioned in that round. These resources are not eligible for double usage in the following round. Additionally, if FSP places a low bid, its profit gets affected. As a result, there is a trade-off between profit and a loss from unsold resources. This problem has been addressed in our work.

We propose a competitive bidding strategy (CompBid) in this paper to maximize the benefit of competition among IoSPs and FSPs. QoS-based Task Allocation and Scheduling (QoTAS) is proposed to maximize resource utilization and profit of all market agents (IoSPs/FSPs/CSP) while taking perishability of resources into account at FSPs. A remote patient monitoring (Verma and Sood, 2018) application is used to compare QoTAS performance to that of previous studies DPDA (Joshi and Srivastava, 2020) and MADA (Peng et al., 2020).

The structure of this paper is as follows:Section 2 discusses related work. The system model and problem statement are explained in Section 3. The task allocation algorithm is explained in Section 4. Section 5 discusses the simulation results, and Section 6 discusses the conclusions.

2 RELATED WORK

In a distributed Cloud-Fog-IoT architecture, IoSPs, FSPs, and CSPs share resources by purchasing them from one another. Each has its own utility function based on the cost and payment received. When considering task allocation using both QoS requirements and monetary costs, the literature divides this problem into two subproblems. These are the allocation and price determination problems. The allocation problem meets QoS requirements, while the price determination problem seeks to maximize profit for all market agents such as FSP, IoSP and CSP. Literature is divided into three categories depending on the agents' bidding strategies: forward auction-based allocation, reverse auction-based allocation, and double auction-based allocation.

Forward Auction-Based Allocation: In a forward auction, buyers submit bids, and the seller allocates resources to the highest paying buyer. (Zu et al., 2019) proposed an ascending bid auction for task offloading problem in fog computing using KKT conditions. (Zhang et al., 2019) presented a Parking Vehicle Assistance (PVA) in VANET using forward auction. Forward auction is appropriate when there are many competing buyers but only one seller or when there is no competition among sellers. In our scenario, however, multiple sellers compete with one another to sell their resources for the highest possible profit. Competitive bidding strategy is proposed in this work.

Reverse Auction-Based Allocation: In a reverse auction, buyers choose the seller with the lowest asking price after competing sellers compete to sell their resources. Reverse auction was utilized by (Guo et al., 2020) to select the target fog nodes and convey the data from the source to the destination fog nodes. Minimizing overall energy use was the goal. For the IoT fog, (Aggarwal et al., 2021) suggest reverse auction-based resource allocation, which also satisfies the QoS criteria in IoT applications. This technique only allows service providers to submit bids. As a result, the seller is unable to profit from the competition among buyers.

Double Auction-Based Allocation: Buyers and sellers submit bids simultaneously in a double auction process, and the auctioneer matches the right buyers and sellers by employing various methods of price determination. (Joshi and Srivastava, 2020) proposed a VCG-inspired price determination algorithm. However, in some cases, VCG may result in negative utility for the broker. CSP serves as a broker in our case. As a result, using CSP may have a negative impact. (Peng et al., 2020) has proposed a resource allocation method that creates a bipartite graph of users and

fog nodes. Edges in the graph are weighted based on QoS requirements. Allocation is then performed using maximum matching of a bipartite graph. The extended McAfee (Yang et al., 2011) method finds the final price for buyers and sellers. Matching is used to find allocation, so an FSP can serve only one user, even if some of its resources can satisfy the needs of other users. In an actual situation, one FSP can satisfy more than one IoSP.

All of the studies discussed thus far allocate all of the resources required by the task in a single round, resulting in a scarcity of resources in that round, which can only satisfy a limited number of tasks. Furthermore, none of them have taken into account the perishable nature of resources when allocating resources. (Miyashita, 2014) and (Safianowska et al., 2017) have proposed a bidding strategy for perishable resource sellers. The allocation algorithm in both of these works lacks a unique allocation technique for allocating maximum resources to avoid loss due to perishability. To avoid loss due to perishability, our algorithm employs a novel resource allocation technique.

Furthermore, bidding strategy is an important component of the auction mechanism. In a double auction mechanism, the strategy must converge to a point where both parties make the most profit while meeting each other's needs.(Chowdhury et al., 2018) and (Thavikulwat and Pillutla, 2008) have proposed a bidding strategy for periodic double auctions where bids are cleared at regular intervals. It is appropriate for our architecture. However, in this bidding strategy, they are capable of wasting a few initial rounds in which the buyer may not receive any resources and the seller may not sell anything in order to find the equilibrium point. However, because our resources are perishable, we must ensure that resources are allocated from the first round itself. To the best of our knowledge, no bidding strategy is proposed for the fog market.

3 PROBLEM FORMULATION

We consider a three-tier architecture,in which there are N IoSPs in the IoT layer, , M FSPs in the Fog layer, and one CSP in the Cloud layer. FSPs communicate with one another via a mesh network. It is assumed that IoSPs receive virtualized services from FSPs (Zhang et al., 2017). As a result, IoSPs can only connect to FSPs via CSP. Also, time is divided into rounds. Allocation occurs at the start of each round. IoSPs and FSPs register with CSP, as shown in Figure 2. CSP sends summary report of the previous round to all registered IoSPs. When the IoSP requires resources, it sends a request to the CSP. A single IoSP can send multiple task requests. IoSPs send their bid for the task, CSP communicates with FSPs and assigns the best FSP to IoSP. IoSP sends input data to FSP, who performs the required task and returns the results to IoSP. Following task completion, IoSP submits a QoS report; CSP calculates the final payment for FSPs and IoSPs.

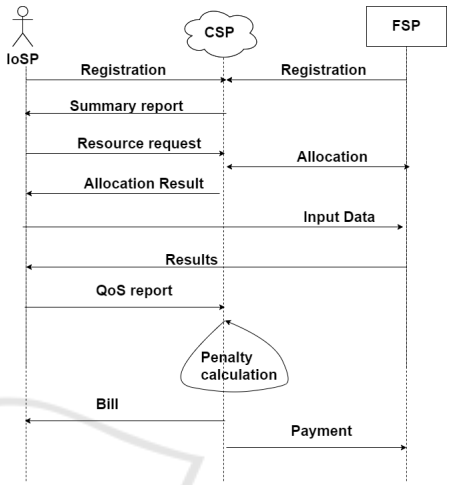


Figure 2: System model.

Resources are represented as Virtual Machines (VMs). Each VM has a set of fixed resources such as memory, network bandwidth, and computing power. Each FSP provides a varying level of QoS service. When resources are limited, higher QoS level VMs are prioritised. It is assumed that the valuation of VMs becomes zero at the end of each round.

Each FSP registers with the CSP by sending fog information $I_j = \{A_j, b^f_{jy}\}$ where A_j is the number of available VMs at j^{th} FSP and b^f_{jx} is bid for x^{th} QoS level. Because CSP stores this information, it is only necessary to send it once during FSP registration. If IoSP wants to execute a task, it sends task requirements to CSP at the beginning of the round. task requirement of i^{th} IoSP for k^{th} task contains $Q_{i,k} = \{\lambda_{i,k}, d_{i,k}, D_{i,k}, s_{i,k}\}$ where $\lambda_{i,k}$ is required number of VMs, $d_{i,k}$ is deadline,$D_{i,k}$ is data size, $s_{i,k}$ is required QoS level at fog devices. At the beginning of each round t CSP sends a summary report (S^t). Using this report IoSP calculate its bid $b^u_{i,k}$ using competitive bidding strategy(CompBid) proposed in the next section and send it to CSP.

1. Maximization of IoSPs utility: IoSP wishes to maximize its utility by adjusting the price paid to CSP. Constraint (1.1) indicates that the task must be completed before the deadline. Con-

straint (1.2) states that each task can only be assigned to one FSP, implying that the task cannot be divided into multiple sub-tasks. (1.3) denotes that each IoSP pays less than its valuation, resulting in a positive utility.

Subproblem-1:	Maximization of IoSP utility	
$\arg\max_{r^u_{i,k}}$	$\sum_{i=1}^{N} U_i^h$	(1.0)
Subject to	$T_{i,k} \leq d_{i,k}, \forall i \forall k$	(1.1)
	$\tau_{i,k,j} \in \{0,1\}$	(1.2)
	$r^u_{i,k} < v_{i,k}, \forall i$	(1.3)

2. Maximization of FSP utility: Each FSP seeks to maximize utility by varying the price and number of allocated VMs (a_j). Constraint (2.1) states that FSP may not allocate more VMs than are available. Constraint (2.2) states that each FSP must be priced higher than its maintenance cost in order to be utility positive.

Subproblem-2:	Maximization of FSP utility	
$\arg\max_{r^f_j, a_j}$	$\sum_{j=1}^{M} U_j^f$	(2.0)
Subject to	$a_j \leq A_j, \forall j$	(2.1)
	$r^f_j > E_j, \forall j$	(2.2)

3. Maximization of CSP utility: CSP wants to maximize its utility by matching IoSPs and FSPs and determining their prices. The requirement in constraint (3.1) states that IoSPs' payments must exceed the price that must be paid to FSPs. Thus, its utility is positive.

Subproblem-3:	Maximization of CSP utility	
$\arg\max_{\tau_{i,k,j}, r^u_{i,k}, r^f_j}$	U^c	(3.0)
Subject to	$\sum_{i=1}^{N} r^u_{i,k} \geq \sum_{j=1}^{M} r^f_j$	(3.1)

These subproblems are interdependent. IoSPs want to maximize their utility by lowering the price to be paid. However, because of this, it may not receive any resources at all. Furthermore, the utility of CSP and FSP may decrease as they receive less payment from IoSP. FSP wishes to maximize its utility by charging higher fees, which may reduce its total allocated resources as well as IoSP and CSP utility. This is an example of a joint optimization problem. It is NP-hard to maximize the utility of all three nodes at the same time(Zhang et al., 2017). Furthermore, perishable nature of cloud-fog resources makes this problem more difficult.

4 PROPOSED SOLUTION

We propose a heuristic-based solution that has two components namely Competitive bidding (CompBid) strategy and QoS-based Task Allocation and Scheduling (QoTAS).

4.1 Competitive Bidding (CompBid)

To maximize service providers' benefits as a result of competition among IoSPs, an IoSP's bid must increase in conjunction with an increase in the demand supply ratio, also known as normalised load. To put this logic into action, we design a bidding strategy in which CSP shares a summary report at the start of each round (t). Which contains $S^t = \{n^{t-1}, c_y^{t-1}, n^t\}$. n^t represents normalized load of t^{th} round and c_y^{t-1} is average VM price of the previous round for y^{th} QoS level. Each IoSP has a summary report published by CSP as an input, as well as its own aggressiveness factor (β_i). Each IoSP determines its own bid for QoS level y using

$$b^u_{i,k} = \min(B_0 + \beta_i \frac{n^{t-1}}{n^t} c_y^{t-1}, v_{i,k}) \quad (1)$$

Here, B_0 is the base price, which should be equal to or greater than the VM's maintenance cost. β_i can be any value between 0 and 1. The greater the value of β_i, the more aggressive the IoSP and the higher the bid.

4.2 QoS-Based Task Allocation and Scheduling (QoTAS)

We propose a double auction-based algorithm for QoS-based Task Allocation and Scheduling (QoTAS) in commercial three-tier Architecture. Algorithm 1 depicts the steps of the QoTAS algorithm. In the first step, we identify FSP that meets QoS requirements for each task. The final winner and price are determined in the second step based on the bids of ISPs and FSPs. Because VMs are perishable resources, in the third step, if there are tasks with eligible FSPs based on QoS requirements that did not win due to a lower bid, we assign VMs of those FSPs to the IoSPs with a penalty.

Algorithm 1: QoS-based Task Allocation and Scheduling (QoTAS).
1: Find eligible FSP for each task(Algorithm 2) 2: Price and final winner determination (Algorithm 3) 3: Allocation of remaining VMs (Algorithm 4)

4.2.1 Find Eligible FSP for Each Task

In this algorithm, we try to find an eligible FSP for each task that has the minimum service delay for the task. We also schedule VMs based on resource scarcity on each FSP for load balancing and efficient VM utilization. In the first step, we identify candidate FSPs for each task. Candidate FSP is an FSP who is capable of completing tasks before the deadline. We may receive a more than one candidate FSPs for each task. Then, using Algorithm 2, we filter them to find the most eligible FSP for each task. To distribute VMs with load balancing so that each FSP receives an equivalent number of tasks. It first computes each FSP's resource scarcity using Eq. (2), which is the ratio of VM demand based on the candidate FSP list to total available VMs on a specific FSP.

$$r_j^s = \frac{\sum_{i=1}^{N}\sum_{k=1}^{X} w_{i,j,k} q_{i,k}}{A_j} \tag{2}$$

Where $w_{i,j,k}$ is one if j^{th} FSP is candidate FSP for k^{th} task of i^{th} IoSP. The minimum penalty $p_{i,k}^{min}$ for a service delay of $d_{i,k+1}$ is calculated for all tasks. It is a penalty if this task is completed one unit of time after its deadline. Following that, all FSPs are sorted in ascending order of r_j^s, with the FSPs with the lowest resource scarcity being allocated first. Tasks, on the other hand, are prioritised using a penalty system. The greater the minimum penalty, the more important that task. As a result, for All tasks, the descending order of $p_{i,k}^{min}$ is used.

Each task in the sorted list is compared against each FSP in the sorted list. If $w_{i,j,k} = 1$, task scheduling is done. We provide a task list, fog information, and candidate FSPs for each task as input to Algorithm 2. For each task, the algorithm returns eligible FSP.

4.2.2 Price Determination

Following the identification of eligible FSPs for each task, we must determine the final winners and the final price to be paid by ISPs, as well as the final payment to be received by FSPs based on their bid. For double auctions, there are numerous winner determination and pricing methods available. We map the requirement of our problem statement with terms of auction theory to find the best method for our scenario; from that exercise it is concluded that, we need Individual rationality (IR), Weekly budget baance (WBB), Truthull (TF) and Allocative efficient (AE) pricing mechanism. We compared k-DA, VCG, Trade reduction, McAfee and extended McAfee techniques and concluded that extended McAfee (Peng et al., 2020) is most suitable pricing mechanism for our model.

Algorithm 2: Find eligible FSP.

Input: Task list, Fog info(I), Candidate list
Output: Eligible Task-FSP pair
for Each FSP in Fog info **do**
  Calculate r_j^s using Eq. (2)
Sort FSPs in ascending order of r_j^s
for Each Task **do**
  Calculate $p_{i,k}^{min}$
Sort Task in descending order of $p_{i,k}^{min}$
for Each Task in sorted Task list **do**
  for Each FSP in sorted Fog info **do**
    if FSP is candidate FSP for Task **then**
      if Deadline of task is one round **then**
        Allocate all required VMs in one round
      else if Deadline($d_{i,k}$) is less than r_j^s **then**
        Allocate VMs equally till $d_{i,k}$ rounds
      else
        Allocate VMs equally till r_j^s rounds

However, we have a bid of IoSPs for different QoS levels so we can't not allocate them altogether. Algorithm 3 shows solution to handle it.

Algorithm 3: Winner and price determination.

Create a group of task requests for each QoS level
We have the price of all FSPs for each QoS level
Apply extended McAfee for each group and find winners and price

4.2.3 Allocation of Remaining VMs

We may not keep VMs idle if IoSPs' bid is less than the required cost since they are perishable. However, if we allocate the same VMs at a lower price, users may always bid lower to maximize their utility. In response to this situation, we propose algorithm 4 where we are allocating lesser VMs than demanded and QoS satisaction is not guaranteed.

Algorithm 4: Allocation of remaining VMs.

Input: Task list, Fog info, Eligible Task FSP pair, Winner pairs
Output: Updated winner pairs
for Each Task in Task list **do**
  for Each FSP in Fog info **do**
    if FSP is eligible for Task but not winner **then**
      Declare that Task FSP pair as winner
      Price to be paid is equal to bid of IoSP
      Allocate γ percentage of required VMs with QoS level one

5 SIMULATION AND RESULTS

A remote patient monitoring case study is used to compare the performance of QoTAS in truthful bidding and CompBid strategy with Deadline and Priority enabled Double Auction (DPDA) (Joshi and Srivastava, 2020) and Multi Attribute Double Auction (MADA) (Peng et al., 2020).

5.1 Simulation Setup

There are four types of tasks in remote patient monitoring. Namely, client module task, pre-processing task, abnormality task, and prediction task. Out of that client module task, pre-processing task, abnormality task are processed on fog nodes (Mahmud et al., 2022). This task request can be generated by a variety of patients. The monitoring period and deadline for processing collected data differs depending on the criticality of the patient (Kumar et al., 2008). We assume that each patient's monitoring period, also known as task generation period, is a multiple of the auction period. As a result, at the start of each auction round, requests from all patients are received.

We implement a scenario in which there are 30 patients, 6 FSPs and one CSP. We consider two tasks here, preprocessing and abnormality detection, because they are expected to be performed on fog. The resource requirements for both of these tasks are listed in (Mahmud et al., 2022). Total simulation time is 500s and allocation is performed at every 30s. Simulation parameters are shown in Table 1. Here,λ_{ik}^{1} represents VM requirement of preprocessing task and λ_{ik}^{2} represents VM requirement of abnormality detection task (Mahmud et al., 2022)

Table 1: Simulation parameters.

Parameter	Value
A_j	30
E_j	N(250,50)
$v_{i,k}$	N(500,100)
B_0	N(250,50)
β_i	N(0,1)
λ_{ik}^{1}	4
λ_{ik}^{2}	5

The patients are classified in four groups here based on their criticality(sensitivity) as shown in Table 2.

Table 2: Patient information.

Class	% of patient	Periodicity	Deadline	Sensitivity
0	10%	30s	10s	3
1	20%	60s	30s	3
2	30%	120s	60s	2
3	40%	240s	90s	1

The delay between the FSP and the ISP is modelled in Netsim by simulating real-time scenarios like congestion, link down, and packet loss, and then used it in Python code for allocation. After this setup, we have compare the following parameters in the allocation mechanism:

1. Task Allocation Ratio(TA): It is ratio of number of tasks successfully executed and number of task requests accepted by CSP.
2. Resource Utilization (RU): It is the ratio of total VMs allocated and total VMs available.
3. Average User Utility(UU): It is the ratio of average IoSP utility per task and task valuation for all assigned tasks.
4. System Utility (SU): It is the sum of the average FSP utility(FU) and the average CSP utility(CU). Where FU is the ratio of FSP utility per task to task maintenance cost. CU is the ratio of the CSP utility per task and the IoSP valuation for that task.

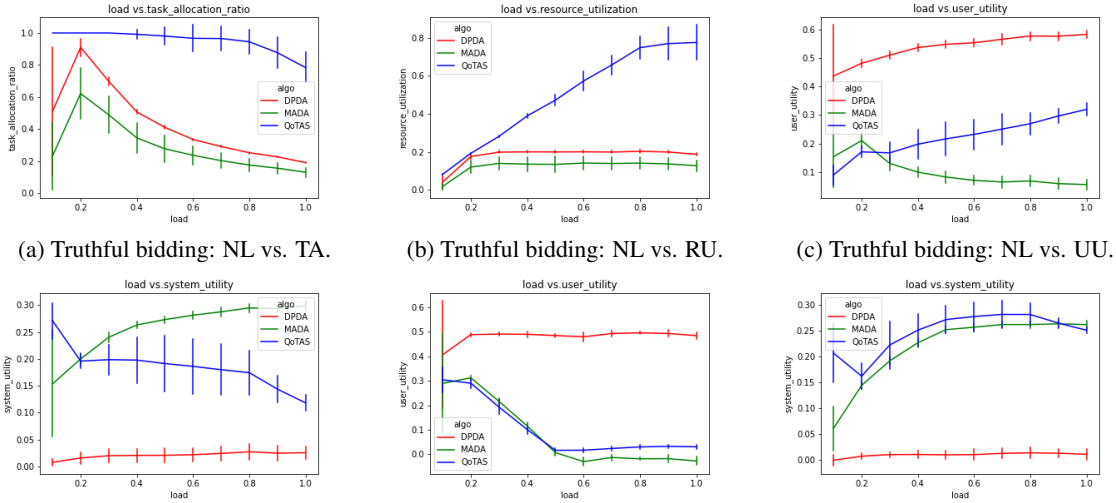
5.2 Results

We measure QoTAS performance using two bidding strategies: truthful bidding and competitive bidding. Normalized load (NL) is changed from 0 to 1 with step size 0.1.

5.2.1 Truthful Biding Strategy

We assume that FSP and IoSP both bid their true task valuations. In this bidding strategy, agents do not consider market scenarios such as resource demand and other agents' bids. As shown in Figure 3(a), MADA's TA is the lowest because it uses a matching algorithm for VM allocation, allowing it to allocate only one task per FSP. Also, because of that its TA decreases as NL increase. DPDA has more TA than MADA because each FSP can be assigned multiple tasks. QoTAS has the highest TA because it assigns resources to tasks in multiple rounds if necessary. It also employs extended McAffee, which allocates the maximum number of FSP and IoSP pairs. Furthermore, QoTAS allocates remaining resources at a low cost as part of perishable allocation, so QoTAS has the highest TA. Overall, QoTAS has 54% more TA than DPDA and 69% more than MADA.

Because more resources are used when more tasks are assigned, the resource utilization (RU) of QoTAS is maximum and the RU of MADA is minimum as shown in Figure 3(b). Also, for MADA, RU remains constant since it always allocated constant number of tasks which is equal to number of FSPs. Overall, QoTAS has 63% more RU than DPDA and 75% more than MADA.



(a) Truthful bidding: NL vs. TA. (b) Truthful bidding: NL vs. RU. (c) Truthful bidding: NL vs. UU.

(d) Truthful bidding: NL vs. SU. (e) CompBid: NL vs. UU. (f) CompBid: NL vs. SU.

Figure 3: Comparision of QoTAS with DPDA and MADA.

As shown in Figure 3(c), the UU of VCG is the highest because in VCG, buyers pay the seller's ask at the breakeven point. In contrast, in extended McAfee, buyers pay the buyer's bid at the break even point. Because the seller's ask at the breakeven point is always less than the buyer's bid, buyers must pay less in VCG. In our case, IoSPs are the buyers. QoTAS has a higher UU than MADA because it completes more tasks after accepting, avoiding penalties and increasing UU. Oerall, DPDA has 59% more UU than QoTAS and QoTAS has 54% more than MADA.

As shown in Figure 3(d), the SU of DPDA is minimum because it employs VCG, and in VCG auctioneer has negative utility because it accepts less payment from buyers and gives more payment to sellers, necessitating the addition of its own funds. In our case, CSP is an auctioneer, so while DPDA has the highest FSP utility , it has the lowest CSP utility, making its total system utility the lowest of all three. MADA has more SU than QoTAS because, due to perishable allocation in QoTAS, it must sell resources at a lower price than FSP's ask, so CSP must add money from its own pocket to pay FSP. This reduces CU and, eventually, SU. Overall, QoTAS has 89% more SU than DPDA and 26% less than MADA.

5.2.2 Competitive Bidding Strategy

In this section, we use CompBid strategy in which IoSPs increased their bid as resource demand increased. TA and RU has same behaviour as truthful biddding since total number of tasks allocated depends on available resources and QoS requirements.

As illustrated in Figure 3(e), the comparative behaviour of DPDA, MADA, and QoTAS is identical to the Truthull bidding strategy depicted in 3(c). The only difference is that the UU of MADA and QoTAS decreases as load increases because IoSPs must pay more because they are expected to pay the lowest bid of all IoSPs and bid increases as load increases. UU for DPDA remains constant because in VCG IoSP pays seller's ask, which is constant here. Overall, DPDA has 78% more UU than QoTAS and QoTAS has 20% more than MADA.

As illustrated in Figure 3(f), the SU of MADA and QoTAS increases as load increases due to higher IoSP bids. For DPDA, SU remains constant because the bid of IoSP increases as the load increases. This causes CSP to pay more to FSP, lowering CU and increasing FU, but the SU is the sum of both, keeping SU constant. Overall, QoTAS has 90% more SU than DPDA and 12% more than MADA.

As a conclusion, QoTAS increase achieves more TA and RU than DPDA and MADA in both bidding strategy. Also, it has more SU than DPDA and more UU than MADA in both truthful and CompBid strategy. Also, it is observed that CompBid strategy increases SU of QoTAS by 25% with some reduction in UU.

6 CONCLUSION

Tasks with deadlines are generated by IoT applications. This task can be handled by cloud-fog service providers. Task allocation and scheduling are difficult in commercial cloud and fog architecture. To solve this problem, a multi-attribute double auction is used. The perishable nature of cloud-fog VMs motivates

allocation of the maximum number of VMs in each round. The algorithm for allocating the remaining VMs after the auction is proposed. To take advantage of competition among users and service providers, a competitive bidding strategy is proposed. To compare QoTAS with existing studies MADA (Peng et al., 2020) and DPDA (Joshi and Srivastava, 2020) remote patient monitoring tasks are used. In both the truthful and competitive bidding scenarios, QoTAS outperforms DPDA and MADA in both bidding strategies. It has more 89% SU than DPDA and more 54% UU than MADA. Furthermore, the CompBid strategy increases SU of QoTAS by 25% while decreasing UU.

REFERENCES

(2022). IoT data. https://www.businesswire.com/news/home/20190618005012/en/The-Growth-in-Connected-IoT-Devices-is-Expected-to-Generate-79.4ZB-of-Data-in-2025-According-to-a-New-IDC-Forecast. [Online; accessed 18-03-2022].

Aggarwal, A., Kumar, N., Vidyarthi, D. P., and Buyya, R. (2021). Fog-integrated cloud architecture enabled multi-attribute combinatorial reverse auctioning framework. *Simulation Modelling Practice and Theory*, 109:102307.

Baranwal, G., Kumar, D., Raza, Z., and Vidyarthi, D. P. (2018). *Auction based resource provisioning in cloud computing*. Springer.

Chowdhury, M. M. P., Kiekintveld, C., Tran, S., and Yeoh, W. (2018). Bidding in periodic double auctions using heuristics and dynamic monte carlo tree search. In *IJCAI*, pages 166–172.

Guo, Y., Saito, T., Oma, R., Nakamura, S., Enokido, T., and Takizawa, M. (2020). Distributed approach to fog computing with auction method. In *International Conference on Advanced Information Networking and Applications*, pages 268–275. Springer.

Jin, A.-L., Song, W., Wang, P., Niyato, D., and Ju, P. (2015). Auction mechanisms toward efficient resource sharing for cloudlets in mobile cloud computing. *IEEE Transactions on Services Computing*, 9(6):895–909.

Joshi, N. and Srivastava, S. (2019). Task allocation in three tier fog iot architecture for patient monitoring system using stackelberg game and matching algorithm. In *2019 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pages 1–6. IEEE.

Joshi, N. and Srivastava, S. (2020). Auction-based deadline and priority enabled resource allocation in fog-iot architecture. In *2020 Futuristic Trends in Network and Communication Technologies(FTNCT)*. Springer.

Kumar, S., Kambhatla, K., Hu, F., Lifson, M., and Xiao, Y. (2008). Ubiquitous computing for remote cardiac patient monitoring: a survey. *International journal of telemedicine and applications*, 2008.

Mahmud, R., Pallewatta, S., Goudarzi, M., and Buyya, R. (2022). Ifogsim2: An extended ifogsim simulator for mobility, clustering, and microservice management in edge and fog computing environments. *Journal of Systems and Software*, 190:111351.

Miyashita, K. (2014). Online double auction mechanism for perishable goods. *Electronic Commerce Research and Applications*, 13(5):355–367.

Peng, X., Ota, K., and Dong, M. (2020). Multiattribute-based double auction toward resource allocation in vehicular fog computing. *IEEE Internet of Things Journal*, 7(4):3094–3103.

Safianowska, M. B., Gdowski, R., and Huang, C. (2017). Preventing market collapse in auctions for perishable internet of things resources. *International Journal of Distributed Sensor Networks*, 13(11):1550147717743693.

Suh, M.-k., Chen, C.-A., Woodbridge, J., Tu, M. K., Kim, J. I., Nahapetian, A., Evangelista, L. S., and Sarrafzadeh, M. (2011). A remote patient monitoring system for congestive heart failure. *Journal of medical systems*, 35(5):1165–1179.

Thavikulwat, P. and Pillutla, S. (2008). Pure and mixed strategies for programmed trading in a periodic double auction. *International Journal of Applied Decision Sciences*, 1(3):338–358.

Verma, P. and Sood, S. K. (2018). Fog assisted-iot enabled patient health monitoring in smart homes. *IEEE Internet of Things Journal*.

Weinman, J. (2017). Fogonomics — the strategic, economic, and financial aspects of the cloud. In *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, volume 1, pages 705–705.

Yang, D., Fang, X., and Xue, G. (2011). Truthful auction for cooperative communications. In *Proceedings of the Twelfth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 1–10.

Zhang, H., Xiao, Y., Bu, S., Niyato, D., Yu, F. R., and Han, Z. (2017). Computing resource allocation in three-tier iot fog networks: A joint optimization approach combining stackelberg game and matching. *IEEE Internet of Things Journal*, 4(5):1204–1215.

Zhang, Y., Wang, C.-Y., and Wei, H.-Y. (2019). Parking reservation auction for parked vehicle assistance in vehicular fog computing. *IEEE Transactions on Vehicular Technology*, 68(4):3126–3139.

Zu, Y., Shen, F., Yan, F., Yang, Y., Zhang, Y., Bu, Z., and Shen, L. (2019). 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)*, pages 1–6. IEEE.