Truthful Multi-resource Transaction Mechanism for P2P Task Offloading based on Edge Computing

Weifeng Lu, Shitao Zhang, Jia Xu, *Member, IEEE*, Dejun Yang, *Senior Member, IEEE*, and Lijie Xu

Abstract*—Peer-to-Peer (P2P) resource sharing promotes local resource-hungry task offloading to other mobile devices and balances the resource consumption between mobile devices. Most of existing P2P task offloading systems aims to solve the resource sharing between one pair exclusively without considering the cost of resource supply and the strategic behaviors of mobile users. In this paper, we propose two user models for the P2P task offloading system: honest user model and strategy user model. For the honest user model, we formulate the resource allocation maximization problem with latency and energy consumption constraints as an Integer Linear Programming. We show that the solution for honest user model can output 189% resource transactions of that for the strategic users. For the strategy user model, we propose a double auction-based P2P task offloading system, and design a truthful multi-resource transaction mechanism to maximize the number of resource transactions. We first group the mobile users based on the connected components to improve the efficiency of double auction. Then we utilize the McAfee Double Auction to price the resource transactions. Finally, we split each winning mobile user of double auction into multiple virtual mobile users, and use the matching approach to calculate the resource allocation. Through both rigorous theoretical analysis and extensive simulations, we demonstrate that the designed multi-resource transaction mechanism satisfies the desirable properties of computational efficiency, individual rationality, budget balance, truthfulness for resource request/supply, and general truthfulness for bid/ask price.

***Index Terms*—resource allocation, double auction, edge computing, task offloading, matching.**

W. Lu, S. Zhang, J. Xu and L. Xu are with the Jiangsu Key Laboratory of Big Data Security and Intelligent Processing, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu 210023, China. (e-mail: luwf@njupt.edu.cn, 1218043329@njupt.edu.cn, xujia@njupt.edu.cn, ljxu@njupt.edu.cn).

D. Yang. is with Colorado School of Mines, Golden, CO 80401. (e-mail: djyang@mines.edu.)

Corresponding author: Jia Xu

This work has been supported in part by the NSFC (No. 61872193, 61872191, 62072254), NSF (No. 1717315), and NUPTSF (No. NY220056)

I. Introduction

In the past decade, the popularity of mobile devices has brought a variety of services to the public. At the same time, wireless communication network has an explosive growth. Smartphone has integrated various resources, such as communication resource, computing resource, data resource and storage resource. Although the resource capacity and hardware technology of mobile devices have significantly improved in recent years, it still cannot meet the requirements of mobile applications, which become more and more complex and resource-hungry.

The recently emerged task offloading architecture Multi-access Edge Computing (MEC) is a promising paradigm to solve this problem. The main feature of MEC is to use a various types of resources (e.g., computing resource, communication resource, storage resource) of the edge nodes to enable resource-intensive and latency-critical applications at the resource-limited mobile devices [1]. A large number of mobile devices, including smartphones, tablets and laptops, have integrated various resources. The ubiquitous underutilized resources of mobile devices can be utilized to help other mobile devices to perform resource-hungery tasks. Comparing with traditional task offloading systems, such P2P task offloading systems can provide more shareable resources with low price.

Short-distance communication technique is a feasible and efficient way to help resource sharing. There are many kinds of proven short-distance communication technologies so far, including Bluetooth, NFC, WLAN, D2D communication, infrared communication and so on. When a large number of mobile phone users are concentrated in a small area, such as office buildings and apartment buildings, there will be a quantity of devices with insufficient or redundant resources, providing the user base of P2P resource sharing. For example, a device running a game can offer communication resource but requires computation resource, while a device with a downloading task can provide computation resource but needs communication resource. With the P2P resource sharing, the downloading can be completed faster, and the player will have better experience.

With the development of short-distance communication technology, many P2P resource sharing systems have been proposed. However, most of existing work [2, 3, 4] considers that the resource sharing only happens in one pair exclusively, even if some devices can provide enough resource to multiple device. Such one-to-one resource sharing model may lead to low resource sharing efficiency. Many-to-many resource sharing model is practicel and efficient. For example, [5] proposes a cooperative video processing system, where the video can be divided into video chunks. Then the system sends the video chunks to different devices for analysis.

Another special problem in P2P resource sharing is that the communication distance of the device is limited. In practice, the devices can be at anywhere, so they are not fully connected. In past studies, this problem was rarely considered. For examples, *Chen et al.* [6] focus on the task offloading to devices that can harvest energy and provide computation offloading services to the nearby users. In this model, the author assumed that participants could establish direct communication connections between others. In [7], all the users are assumed in close proximity of each other.

From the perspective of resource allocation mechanism design, many researches [8, 9, 10] did not take into account the strategic behavior of users, which can seriously hinder the potential collaboration of users. For example, in [11], a multi-cell, multi-server MEC system is considered and each base station is equipped with a MEC server that assists mobile users by providing computation offloading services. In this system, in order to execute the allocation algorithm, MEC servers and users need to submit private information. However, this algorithm does not take into account that the MEC servers and users may lie on the private information. In practice, the users usually tend to lie about the private information in order to improve their utilities. Therefore, the designed resource sharing mechanism should discourage the strategic behaviors. *STrategy-proof double Auctions for multi-cloud, multi-tenant bandwidth Reservation* (*STAR*) [12] is an open market of cloud bandwidth reservation using double auction to match cloud tenant with cloud providers. *STAR* contains two auction mechanisms, and the second mechanism named *STAR-Padding* can be used in the general scenario, in which the bandwidth capacities of the cloud providers can be different, and the tenants demands are divisible. [13] proposes a truthful double auction mechanism for sensing task allocation, where there are multiple tasks in the crowdsensing system. In [14], authors propose a multi-attribute-based double auction mechanism in vehicular fog computing, which considers both the price and non-price attributes and can ensure the truthfulness. In summary, the above three resource transaction mechanisms cannot be used directly in our P2P task offloading system due to the connectivity constraint and request/supply diversity. Futher, different from the truthful allocation algorithm proposed in [15, 16], truthfulness for both bid/ask price and resource request/supply simultaneously is desired in our system.

Our study focuses on designing the task offloading mechanisms that enable mobile users (MUs) to share their resources flexibly by addressing the above-mentioned problems. In our system, we consider that each MU requests some types of resources, and the demand for each type of resource is different. Additionally, the task is divisible and can be offloaded to multiple MUs within the communication range.

The problem of designing multi-resource transaction mechanism for P2P task offloading system is very challenging. First, the resource sharing of MUs typically incurs cost, such as energy consumption and data usage. The MUs may take a strategic behavior by submitting dishonest bid/ask price or resource request/supply to maximize their utilities. The designed mechanism should stimulate the truth-telling of private information of MUs. Second, each MU may request different number of resources. This means we cannot straightforwardly use the traditional *Single-unit McAfee Double Auction* [17], where each seller/buyer only sells/buys one unit of items. Moreover, due to the different request/supply of MUs, the matching approach, which is usually used for the resource allocation problem, also cannot be applied directly. Finally, the connectivity of MUs can largely affect the efficiency of resource transaction due to the possible disconnection between the traders.

The main contributions of this paper are as follows:

- We present two system models: honest MU model and strategic MU model. In honest MU model, we assume that there is no cost to the resource provider, and all MUs submit their resource request/supply honestly. In strategic MU model, there is a private cost of resource provider, and the MUs can behave strategically by submitting dishonest bid/ask price and resource request/supply.
- We formulate the allocation maximization problem for honest MUs as an Integer Linear Programming (ILP) and obtain the optimal solution through the integer programming problem solver.
- We further formulate the transaction maximization problem for the strategic MUs. We propose a *Truthful Multi-resource Transaction Mechanism (TMTM)*, which integrates *McAfee Double Auction* and matching approach, to maximize the number of resource transactions.
- We show that the designed truthful multi-resource transaction mechanism satisfies the desirable properties of computational efficiency, individual rationality, budget balance, value/cost-truthfulness, and request/supply-truthfulness.

The rest of the paper is organized as follows. Section II formulates and solves allocation maximization problem for honest MUs. Section III formulates the system model for strategic MUs and lists some desirable properties. Section IV presents the design rationale of *TMTM*. Section V presents the detailed design and the analysis of *TMTM*. Performance evaluation is presented in Section VI. Section VII discusses the implementation problems of *TMTM*. Section VIII concludes this paper.

II. Transaction Maximization for Honest MU Model

We consider a P2P task offloading system consisting of a BS and a set of MUs. Due to the limit of resources, each MU has a task that is expected to be completed through resource sharing. Each task can be divided into multiple subtasks, each subtask requires one type of resource, and the subtasks can be offloaded to other neighboring MUs.

We denote $\mathcal{N} = \{1, 2, ..., N\}$ as the set of MUs. The types of resources are denoted as $\mathcal{K} = \{1, 2, ..., K\}$. For each resource $k \in \mathcal{K}$ and each MU $i \in \mathcal{N}$, let τ_i^k be the task of MU i requesting resource k. Let r_i^k and s_i^k be the request and supply of MU i for resource k, respectively. Note that no MU can request and supply for the same resource simultaneously. Both r_i^k and s_i^k are normalized by the resource unit, i.e., $r_i^k, s_i^k \in \mathbb{N}$. Let b_i^k and a_i^k be the unit value/cost of MU i to request/supply resource k of one unit, respectively.

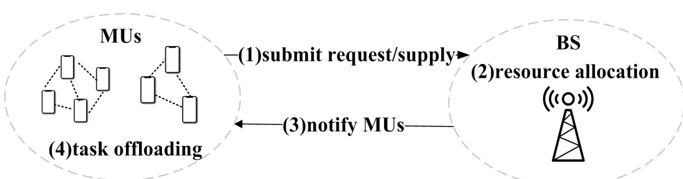


Fig. 1. Task offloading system under honest MU model

As illustrated by Fig. 1, each MU i first submits r_i^k, s_i^k, b_i^k and a_i^k for every resource k to BS. In this section, we consider that each MU i is honest, and it always submits the real values of r_i^k , s_i^k, b_i^k and a_i^k to the BS. Then the BS calculates the resource allocation $\mathbf{Q}^{N \times N \times K}$, where $Q_{i,j}^k$ is the number of resource k MU i obtained from MU j. The BS notifies the MUs of the resource allocation. Finally, the MUs offload the tasks based on the resource allocation.

We give an illustrative example of P2P multi-resource allocation. Let $k = 1, 2, 3$ indicate computation resource, communication resource and storage resource, respectively. The computation resource is the CPU cycles, which may be required by the computation sensitive tasks. The communication resource is the data usage, which may be required by the multimedia data uploading tasks. The storage resource may be required by the urgent data storage tasks. For example, suppose that a sensor node would die soon due to the energy depletion. Then its sensed data needs to be transferred to other sensor nodes urgently. The sensed data is dividable and can be stored in multiple sensor nodes since it will be uploaded to the cloud server ultimately. Note that the resource request and resource supply are normalized by the resource unit. For example, the resource units of computation resource, communication resource and storage resource are 0.1 GHz, 1 KB and 1 KB, respectively. We consider that there are two MUs. MU i requests computation resources of 60 units and communication resources of 50 units, i.e., $r_i^1 = 60, r_i^2 = 50$. Meanwhile, MU i can supply storage resources of 50 units, i.e., $s_i^3 = 50$. MU j can supply computation resources of 40 units and communication resources of 60 units, i.e., $s_j^1 = 40, s_j^2 = 60$. Meanwhile, MU j requests storage resources of 30 units, i.e., $r_j^3 = 30$. Theoretically, we can obtain the maximum resource allocation of 120 units between these two MUs by $Q_{i,j}^1 = 40$, $Q_{i,j}^2 = 50$ and $Q_{j,i}^3 = 30$, where $Q_{i,j}^1 = 40$ represents that MU i obtains computation resources of 40 units from MU j, $Q_{i,j}^2 = 50$ represents that MU i obtains communication resources of 50 units from MU j, and $Q_{j,i}^3 = 30$ represents that MU j obtains storage resources of 30 units from MU i.

We consider that the communication between MUs is based on Orthogonal Frequency Division Multiple Access (OFDMA). The data transmission rate of the link from any MU i to MU j can be given as:

$$R_{i,j} = W_{i,j} log\left(1 + \frac{Pt_i l_{i,j}^{-\beta}}{\sigma^2}\right) \tag{1}$$

where $W_{i,j}$ is the bandwidth of the link from MU i to MU j. Pt_i is the transmission power of MU i. $l_{i,j}$ is the distance between MU i and MU j. β is the path loss exponent. σ^2 is the noise power of the channel.

For any two MUs $i, j \in \mathcal{N}$ and resource $k \in \mathcal{K}$, the unit resource execution time of task τ_i^k by using MU j's resource can be calculated as:

$$T_{i,j}^k = \frac{D_i^k}{R_{i,j}} + TP_{i,j}^k \tag{2}$$

where D_i^k is the unit resource data volume that needs to be offloaded. $D_i^k / R_{i,j}$ represents the unit resource uplink transmission time. $TP_{i,j}^k$ represents the unit resource task processing time, which depends on the type of requested resource.

We omit the downlink transmission time since the data volume of processing results is usually much smaller than that of tasks. Particularly, the tasks requesting communication resource or storage resource only need the acknowledgements, and the downlink transmission time is very small.

Here, we give the unit resource task processing time for computation resource, communication resource and storage resource. Same as the illustrative example given above, let $k = 1, 2, 3$ indicate computation resource, communication resource and storage resource, respectively. Then, unit resource task processing time can be calculated as:

$$TP_{i,j}^{k} = \begin{cases} \frac{D_i^k}{f}, & k=1 \\ \frac{D_i^k}{R_{j,0}}, & k=2 \\ 0, & k=3 \end{cases} \tag{3}$$

where f is the CPU cycle frequency of unit computation resource, e.g., f=0.1 GHz in the illustrative example. $R_{j,0}$ is the data transmission rate of the link from MU j to the BS. $R_{j,0}$ can be calculated through same method given in formula (1). We omit the storage time since it is rather small comparing with computation time and transmission time.

For arbitrary two MUs $i, j \in \mathcal{N}$ and resource $k \in \mathcal{K}$, the unit resource energy consumption of task τ_i^k by using MU j's resource can be calculated as:

$$H_{i,j}^{k} = Pt_i \frac{D_i^k}{R_{i,j}} + HP_{i,j}^{k} \tag{4}$$

where $Pt_i \frac{D_i^k}{R_{i,j}}$ represents the unit resource energy consumption for uploading data volume D_i^k. $HP_{i,j}^{k}$ represents the unit resource energy consumption for task processing, which also depends on the type of requested resource. For same reason, we omit the downloading energy consumption.

The unit resource energy consumption for task processing can be calculated as:

$$HP_{i,j}^{k} = \begin{cases} \alpha f^3 \cdot TP_{i,j}^{k}, & k=1 \\ Pt_j \cdot TP_{i,j}^{k}, & k=2 \\ 0, & k=3 \end{cases} \tag{5}$$

where α is the effective capacitance coefficient of the CPU. We omit the energy consumption of storage since it is rather small comparing with the energy consumption of computation and transmission.

To guarantee the offloading efficiency and low energy consumption, we set the latency constraint and the energy consumption constraint for each possible resource allocation. We set the threshold of unit resource execution time T^k and the threshold of unit resource energy consumption H^k for each resource k. The thresholds depend on the quality of service provided by the P2P task offloading systems.

For any MU i, the resource of k that i is provided can be calculated as $\sum_{j \in \mathcal{N}, j \neq i} Q_{i,j}^{k}$. Accordingly, the total resource supply from all MUs is:

$$\sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}, j \neq i} Q_{i,j}^{k} \tag{6}$$

Note that the resource sharing only happens between two MUs who are close to each other. We denote the connectivity of all MUs as connectivity matrix $\mathbf{d}$, where $d_{i,j} = 1$ if MU i and MU j are in the range of their communications, and $d_{i,j} = 0$ otherwise. Specifically, $d_{i,j} = 0$ if $i = j$. The connectivity matrix can be obtained through BS location service.

The objective is maximizing the resource allocation of the P2P task offloading system, which can be formalized as:

$$\max \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}, j \neq i} Q_{i,j}^{k} \tag{7a}$$

$$s.t.: \sum_{j \in \mathcal{N}, j \neq i} Q_{i,j}^{k} \leq r_i^k, \forall i \in \mathcal{N}, k \in \mathcal{K} \tag{7b}$$

$$0 \leq \sum_{j \in \mathcal{N}, j \neq i} Q_{j,i}^{k} \leq s_i^k, \forall i \in \mathcal{N}, k \in \mathcal{K} \tag{7c}$$

$$\sum_{k \in \mathcal{K}} (b_i^k \sum_{j \in \mathcal{N}, j \neq i} Q_{i,j}^{k} - a_i^k \sum_{j \in \mathcal{N}, j \neq i} Q_{j,i}^{k}) \geq 0, \forall i \in \mathcal{N} \tag{7d}$$

$$Q_{i,j}^{k} \geq 0, \forall i, j \in \mathcal{N}, k \in \mathcal{K} \tag{7e}$$

$$Q_{i,j}^{k} = 0, \text{ if } d_{i,j} = 0, \forall i, j \in \mathcal{N}, k \in \mathcal{K} \tag{7f}$$

$$T_{i,j}^{k} \leq T^k, \quad \forall i, j \in \mathcal{N}, k \in \mathcal{K} \tag{7g}$$

$$H_{i,j}^{k} \leq H^k, \ \forall i, j \in \mathcal{N}, k \in \mathcal{K} \tag{7h}$$

The constraint (7b) means that the number of resource k allocated from other MUs to any MU should not be more than its request. The constraint (7c) means that the number of resource k it provides should not be more than its supply. The constraint (7d) ensures that the total value is not less than the total cost for any MU, i.e., the payoff of each MU is non-negative. The constraint (7e) means that the resources provided by any MU is non-negative. The constraint (7f) ensures that the task offloading only happens between the MUs in proximity. The constraint (7g) means that the unit resource execution time should meet the latency requirement. The constraint (7h) means that the unit resource energy consumption is limited by the energy consumption requirement. We consider that the BS knows the bandwidth of the links between MUs, the transmission power and position of each MU. The unit resource data volume is submitted by the resource requester.

The constraint (7b) is resonable because the redundant resources cannot create value for any MU. On the other hand, we consider that the MUs are not single-minded. In other words, the required resources can be not fully satisfied. This is because of the characteristic of P2P task offloading. Different from offloading tasks to cloud/edge servers, the resources of MUs are usually not sufficient and probably cannot fulfill the all resource requests. Thus, it is more practical to satisfy partial resource requests in the scenario of P2P task offloading.

Despite the quality of services may decreases, satisfying partial resource requests still has many good application prospects. For examples, obtaining as many communication resources as possible to upload as much data as possible to the remote servers, obtaining as many computation resources as possible to obtain as many results as possible and obtaining as many storage resources as possible to store as much data as possible before device shutdown. To improve the quality of services, we maximize the resource allocation for P2P task offloading. Note that the allocation algorithm can be performed periodically, and the MUs can go on requesting the resources in the next allocation round if the resource requests are not fully satisfied in the current allocation round. In addition, the periodic execution makes the allocation algorithm be adaptable to the time-varying environments.

We can see that the above transaction maximization problem is an Integer Linear Programming (ILP) problem, which can be solved by the integer programming problem solvers, e.g., IBM CPLEX Optimizer [18].

III. System Model and Desirable Properties for Strategic MUs

A. System Model

In the previous section, we assume that all MUs submit their request, supply, unit value, and unit cost honestly. In this section, we consider the MUs are selfish and rational individuals. Each MU can behave strategically by submitting a dishonest request, supply, unit value, or unit cost to maximize its utility.

Auction is a powerful tool to design strategy-proof mechanisms for many resource allocation problems, such as spectrum allocation [19] and task allocation for crowdsensing [20, 21, 22]. To address the resource allocation and pricing problem, we model the task offloading system as a double auction.

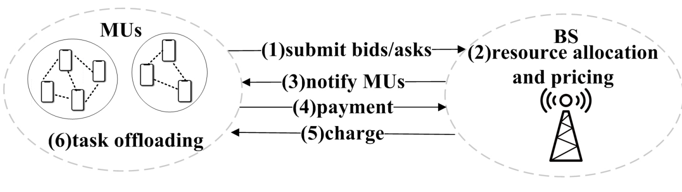


Fig. 2. Double auction based task offloading system.

The definitions of $\mathcal{N}, \mathcal{K}, r_i^k, s_i^k, Q_{i,j}^k, \mathbf{d}, d_{i,j}, D_i^k$ are the same as those in Section II. As illustrated by Fig. 2, we consider an open resource transaction market consisting of a BS and many MUs. For each resource k, if any MU i requests the resource k, it will submit a bid $B_i^k = (r_i^k, D_i^k, b_i^k)$ to the BS, where b_i^k is the unit bid price in auction, i.e., the maximum price it can pay for buying one unit of resource k. Each buyer i of resource k has a real request $\widetilde{r_i^k}$ and value v_i^k , both of which are the private information and known only to buyer i. Accordingly, if any MU i can supply the resource k, it will submit an ask $A_i^k = (s_i^k, a_i^k)$ to the BS, where a_i^k is the unit ask price in auction, i.e., the minimum price it wants to charge for selling one unit of resource k. Each seller i of resource k has a real supply $\widetilde{s_i^k}$ and cost c_i^k, both of which are the private information and known only to seller i.

Given the MU set $\mathcal{N}$, resource set $\mathcal{K}$, connectivity matrix $\mathbf{d}$, bid profile $\mathbf{B}^{N\times K}$, and ask profile $\mathbf{A}^{N\times K}$, the BS calculates the allocation vector $\mathcal{M} = (\mathcal{M}^1, \mathcal{M}^2, ..., \mathcal{M}^K)$, where $\mathcal{M}^k$ is the allocation for resource $k \in \mathcal{K}$, and the payment profile $\mathbf{p} = (\mathbf{p}^1, \mathbf{p}^2, ..., \mathbf{p}^K)$, where $\mathbf{p}^k = (p_1^k, p_2^k, ..., p_N^k)$ is the payment profile for resource $k \in \mathcal{K}$, p_i^k is the payment/charge to MU i for resource k. Then the BS notifies the MUs of the result, gets the payment from the buyers, and pays the charge to the sellers. Finally, the MUs execute the transactions based on the allocation vector.

We define the utility of any seller i as the difference between the total payment and its total real cost for all resources:

$$u_i^s = \sum_{k\in K, s_i^k>0} ((p_i^k - c_i^k) \sum_{j\in N, j\neq i} Q_{j,i}^k) \tag{8}$$

We define the utility of any buyer i as the difference between the total value and its total charge for all resources:

$$u_i^b = \sum_{k\in K, r_i^k>0} ((v_i^k - p_i^k) \sum_{j\in N, j\neq i} Q_{i,j}^k) \tag{9}$$

Thus, the utility of any MU i is:

$$u_i = u_i^s + u_i^b \tag{10}$$

Note that the utility of any MU who is not included in allocation vector $\mathcal{M}$ will be zero since it does not buy or sell any resource in the system, i.e., $u_i = 0$, if $\sum_{k\in\mathcal{K}} \sum_{j\in\mathcal{N}} Q_{i,j}^k + \sum_{k\in\mathcal{K}} \sum_{j\in\mathcal{N}} Q_{j,i}^k = 0$. Moreover, the utility of any MU who cannot supply the resources it claimed will be zero, i.e., $u_i = 0$, if $\sum_{j\in\mathcal{N}} Q_{j,i}^k > \widetilde{s_i^k}, \forall k \in \mathcal{K}$.

We define the utility of BS as the difference between total payment and total charge:

$$u_0 = \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{N}, r_i^k>0} \sum_{j\in\mathcal{N}, s_j^k>0} (p_i^k - p_j^k) Q_{i,j}^k \tag{11}$$

Since we consider that the MUs are selfish and rational individuals, each buyer i can behave strategically by submitting a dishonest request or dishonest unit bid price to maximize its utility. Accordingly, each seller i can behave strategically by submitting a dishonest supply or dishonest unit ask price to maximize its utility.

The objective is maximizing the number of transactions between the MUs, i.e., maximizing the total size of all resource allocations $\sum_{k\in\mathcal{K}} |\mathcal{M}^k|$ under the constraints of latency and energy consumption, where $|\mathcal{M}^k|$ means the number of elements in vector $\mathcal{M}^k$.

B. Desirable Properties

Our objective is to design the multi-resource transaction mechanism satisfying the following four desirable properties:

- **Computational Efficiency:** A multi-resource transaction mechanism is computationally efficient if the allocation vector and the payment profile can be computed in polynomial time.
- **Individual Rationality:** Each MU will have a non-negative utility when bidding its true request/supply and unit bid/ask price, i.e., $u_i \geq 0, \forall i \in \mathcal{N}$.
- **Budget Balance:** The BS will have a non-negative utility, i.e., $u_0 \geq 0$.
- **Truthfulness:** A multi-resource transaction mechanism is request/supply-truthful and value/cost-truthful (called truthful simply) if reporting the real request/supply and value/cost is a weakly dominant strategy for all MUs. In other words, no MU can improve its utility by submitting a false request/supply or value/cost, no matter what others submit. Specifically, we say the multi-resource transaction mechanism is value/cost generally truthful if no MU can improve its utility in expectation by submitting a false value/cost, no matter what others submit.

The importance of the first three properties is obvious, because they together assure the feasibility of the multi-resource transaction mechanism. The fourth property is indispensable for guaranteeing the compatibility. Being truthful, the multi-resource transaction mechanism can eliminate the fear of market manipulation and the overhead of strategizing over others for the MUs.

We list the frequently used notations in TABLE I.

TABLE I
FREQUENTLY USED NOTATIONS

Notation	Description
$\mathcal{N}, \mathcal{K}$	MU set, resource set
N, K	Number of MUs, number of resource
$\mathcal{N}^k$	MU set of resource k
G, g_k	group set, group k
r_i^k, s_i^k	request of MU i for resource k, supply of MU i for resource k
$\widetilde{r_i^k}, \widetilde{s_i^k}$	real request of i for resource k, real supply of i for resource k
a_i^k, b_i^k	unit ask price of i for resource k, unit bid price of i for resource k
c_i^k, v_i^k	cost of i to supply resource k, value of i to obtain resource k
$\mathbf{d}, \mathbf{d}_{i,j}$	connectivity matrix, connectivity between i and j
$\mathbf{d}^k$	connectivity matrix of resource k
$\mathbf{d}_{i,j}^k$	connectivity between i and j of resource k
$\mathbf{Q}^{N\times N\times K}$	resource allocation
$Q_{i,j}^k$	number of resource k MU i obtained from j
τ_i^k	task of MU i requesting reasource k
$R_{i,j}$	date transmission rate of the link from i to j
D_i^k	unit resource data volume of i for resource k
$T_{i,j}^k$	unit resource execution time of task τ_i^k by using j's resource
$TP_{i,j}^k$	unit resource processing time of task τ_i^k by using j's resource
$H_{i,j}^k$	unit resource energy consumption of task τ_i^k by using j's resource
$HP_{i,j}^k$	unit resource processing energy consumption of task τ_i^k by using j's resource
T^k, H^k	latency constraint and energy consumption constraint of resource k
f	CPU cycle frequency of unit computation resource
$W_{i,j}$	bandwidth of the link from i to j
$l_{i,j}$	distance between i and j
A_i^k, B_i^k	ask of i for resource k, bid of i for resource k
$\mathbf{A}^{N\times K}, \mathbf{B}^{N\times K}$	ask profile for resource k, bid profile for resource k
$\mathcal{M}, \mathcal{M}^k$	allocation vector, allocation for resource k
$\mathbf{p}, \mathbf{p}^k$	payment profile, payment profile for resource k
p_i^k	payment/charge to i for resource k
u_i^s, u_i^b	utility of seller i; utility of buyer i
u_i, u_0	utility of MU i, utility of BS
$\mathcal{N}^{k,h,s}$	seller set of resource k in group h
$\mathcal{N}^{k,h,b}$	buyer set of resource k in group h
$E^{k,h,s}$	winning seller set of resource k in group h
$E^{k,h,b}$	winning buyer set of resource k in group h
$VE^{k,h,s}$	virtual seller set of resource k in group h
$VE^{k,h,b}$	virtual buyer set of resource k in group h
φ	maximum supply/request of MUs

IV. Design Rationale of TMTM

In this section, we propose the key technologies used in our truthful multi-resource transaction mechanism. The basic idea of $TMTM$ is executing the double auctions for every resource iteratively. For each resource $k \in \mathcal{K}$, $TMTM$ consists of grouping phase, pricing phase, and allocation phase. Note that $TMTM$ can be conducted periodically to meet the large-scale resource request or dynamic resource request. Here, we give the design rationale of these three phases briefly in the following subsections.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2021.3079258, IEEE Transactions on Vehicular Technology

A. Grouping

In our system, the resource transactions only happen between the MUs within their communication range. The disconnected MUs will result in the inefficiency of double auction. Thus, BS first conducts a preprocessing to remove some disconnected MUs before the double auction. Specifically, for any resource $k \in \mathcal{K}$, the grouping phase complies with the following steps:

1) Remove the MUs who do not request or supply resource k and the corresponding links from the connectivity matrix $\mathbf{d}$. 2) Since the transactions only happen between sellers and buyers, we further remove the links, which connect two sellers or two buyers. 3) In addition, we need to consider the latency constraint and energy consumption constraint of each possible resource transaction. The BS calculates the unit resource execution time according to formula (2) and the unit resource energy consumption according to formula (4) for each link. If any link cannot satisfy the latency constraint or energy consumption constraint, it should be removed. 4) Remove the isolated MUs. 5) Finally, we group the residual MUs based on the connected components of connectivity matrix.

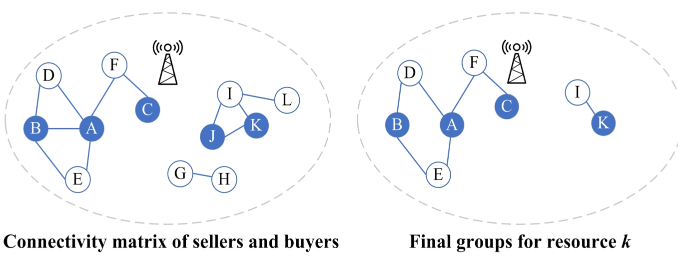


Fig. 3. An example of grouping phase, where the blue nodes represent sellers, the nodes with light color represent buyers. The lines between the nodes represent the links between the MUs. The letters in the nodes represent the IDs of MUs.

As illustrated in Fig. 3, we assume that there are 12 MUs who request or supply. We remove the links $(A, B), (J, K), (I, L)$ and (G, H), which connect two sellers or two buyers. Suppose that the link from MU I to MU J cannot satisfy the energy consumption constraint of resource k. We remove the link (I, J) since the resource transactions on this link are not feasible. Then we remove the corresponding isolated MUs G, H, L and J. Finally, we partition the residual MUs into two groups based on the connected components. As a result, we reduce the number of MUs from 12 to 8 through the grouping phase.

B. Pricing

A straightforward idea is to split each seller/buyer into multiple virtual sellers/buyers with resource request/supply of single unit for each resource $k \in \mathcal{K}$, and apply *Single-unit McAfee Double Auction*, which is a well-known individually rational, budget balanced and value/cost truthful double auction, to decide the winners and payment. Unfortunately, this approach is not request/supply truthful under the system model considered in this paper.

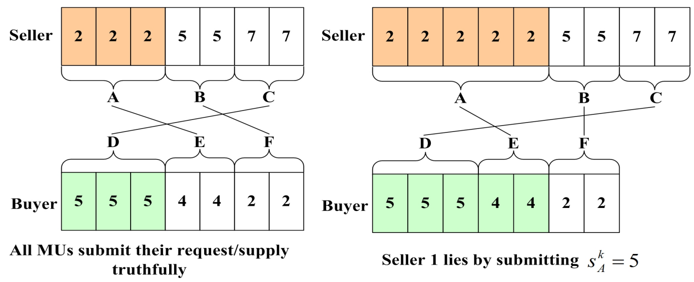


Fig. 4. An example showing the untruthfulness of the single-resource double auction, where each block represents one resource unit, the number inside each block represents the unit ask/bid price. The letters below/above the block represent the MUs. The lines represent the links between the MUs.

We use an example in Fig. 4 to show the statement. In this example, $s_A^k = 3$, $a_A^k = 2$, $s_B^k = 2$, $a_B^k = 5$, $s_C^k = 2$, $a_C^k = 7$, $r_D^k = 3$, $a_D^k = 5$,$r_E^k = 2$, $a_E^k = 4$, $r_F^k = 2$, $a_F^k = 2$. We apply the *Single-unit McAfee Double Auction* to decide the winning sellers/buyers and payment/charge. As a result, the winning seller is A, and the winning buyer is D. Since there is no link between these two MUs, no transaction of resource k happens, and the utility of seller A is zero.

We now consider that seller A lies by submitting $s_A^k = 5$. As a result, the winning seller is A, and the winning buyers are D and E. Since there is a link between seller A and buyer E, seller A can sell 2 units of resource to buyer E, and obtain utility of $(\frac{5+2}{2} - 2) \times 2 = 3$. Note that seller A increases its utility from 0 to 3 by lying about its supply.

The failure of guaranteeing request/supply truthfulness makes *Single-unit McAfee Double Auction* less attractive. In *TMTM*, we conduct the double auction based on the sellers and buyers with resource request/supply of multiple units to price the transactions.

C. Allocation

So far, we have obtained the winning sellers/buyers and their charge/payment through the double auction. Since the sellers/buyers can request/supply different number of resource units, we split each winning seller/buyer into multiple virtual sellers/buyers with resource request/supply of single unit. The problem of maximizing the transactions between the MUs is equivalent to the *MBM (Maximum Bipartite Matching)* problem [23] on the bigraph consisting of virtual sellers and virtual buyers. It is well known that the *MBM* problem can be solved by *FFA (Ford - Fulkerson Algorithm)* [24]. There may be multiple optimal solutions

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2021.3079258, IEEE Transactions on Vehicular Technology

for *MBM* problem, we select one of them randomly as the resource allocation.

V. Design Details of TMTM

In this section, we present the design details of *TMTM*, which consists of three phases: grouping, pricing, and allocation, for the strategic MU model.

A. Mechanism Design

As illustrated by Algorithm 1, $TMTM$ processes each resource $k \in \mathcal{K}$ iteratively. For each resource k, the function **Grouping**$(\cdot)$ return the group set G and the connectivity matrix $\mathbf{d}^k$ of resource k. Based on the group set G, $TMTM$ executes pricing phase and allocation phase through calling function **Transaction**$(\cdot)$, which outputs the allocation $\mathcal{M}^k$ and payment profile $\mathbf{p}^k$ of resource k.

Algorithm 1 : TMTM

Input: MU set $\mathcal{N}$, resource set $\mathcal{K}$, connectivity matrix $\mathbf{d}$, bid profile $\mathbf{B}^{N\times K}$, and ask profile $\mathbf{A}^{N\times K}$

1: $G \leftarrow \emptyset$; $\mathcal{M} \leftarrow \emptyset$;
2: **foreach** $k \in \mathcal{K}$ **do**
3: $\quad\mathbf{d}^k \leftarrow 0$; $\mathcal{M}^k \leftarrow \emptyset$; $\mathbf{p}^k \leftarrow 0$;
4: $\quad(G, \mathbf{d}^k) \leftarrow$ **Grouping**$(\mathcal{N}, k, \mathbf{d}, \mathbf{B}^{N\times K}, \mathbf{A}^{N\times K})$;
5: $\quad(\mathcal{M}^k, \mathbf{p}^k) \leftarrow$ **Transaction**$(G, k, \mathbf{d}^k, \mathbf{B}^{N\times K}, \mathbf{A}^{N\times K})$;
6: **end for**
7: **return** $(\mathcal{M} = (\mathcal{M}^1, \mathcal{M}^2, ..., \mathcal{M}^K)$, $\mathbf{p} = (\mathbf{p}^1, \mathbf{p}^2, ..., \mathbf{p}^K))$;

The grouping phase is illustrated in Algorithm 2. Let $\mathcal{N}^k$ be the MU set of resource k. We first remove the MUs who do not request or supply resource k from $\mathcal{N}^k$ and remove the corresponding links from the connectivity matrix $\mathbf{d}^k$ (Lines 2-9). Afterwards, we further remove the links, which connect two sellers or two buyers (Lines 12-14), the unfeasible links (Lines 15-17) and the corresponding isolated MUs (Line 20). Finally, we group the residual MUs based on the connected components of connectivity matrix $\mathbf{d}^k$ (Line 21).

Next, we present the function **Transaction**$(\cdot)$ illustrated in Algorithm 3. We process each group iteratively. For each group $g_h \in G$, if any MU asks for resource supply, we put it into the seller set $\mathcal{N}^{k,h,s}$. If any MU bids for resource request, we put it into the buyer set $\mathcal{N}^{k,h,b}$ (Lines 3-6). Then, we sort the sellers based on their unit ask prices in nondecreasing order, and sort the buyers based on their unit ask prices in nonincreasing order (Lines 7-8). Let t be the last position for sellers and buyers such that $a_i^k \leq b_i^k$. Then we execute function **Pricing**$(\cdot)$ to determine the winning seller set $E^{k,h,s}$, winning buyer set $E^{k,h,b}$, and payment profile $\mathbf{p}^k$ (Line 10). Afterwards,

Algorithm 2 : Grouping

Input: MU set $\mathcal{N}$, resource k, connectivity matrix $\mathbf{d}$, bid profile $\mathbf{B}^{N\times K}$, and ask profile $\mathbf{A}^{N\times K}$

1: $\mathcal{N}^k \leftarrow \mathcal{N}$; $\mathbf{d}^k \leftarrow \mathbf{d}$;
2: **foreach** $i \in \mathcal{N}$ **do**
3: $\quad$**if** $r_i^k = 0$ **or** $s_i^k = 0$ **then**
4: $\quad\quad\mathcal{N}^k \leftarrow \mathcal{N}^k \backslash \{i\}$;
5: $\quad\quad$**foreach** $j \in \mathcal{N}$ **do**
6: $\quad\quad\quad d_{i,j}^k \leftarrow 0$; $d_{j,i}^k \leftarrow 0$;
7: $\quad\quad$**end for**
8: $\quad$**end if**
9: **end for**
10: **foreach** $i \in \mathcal{N}^k$ **do**
11: $\quad$**foreach** $j \in \mathcal{N}^k$ **do**
12: $\quad\quad$**if**$(r_i^k \neq 0$ **and** $r_j^k \neq 0$ **and** $d_{i,j}^k \neq 0)$ **or** $(s_i^k \neq 0$ **and** $s_j^k \neq 0$ **and** $d_{i,j}^k \neq 0)$ **or** **then**
13: $\quad\quad\quad d_{i,j}^k \leftarrow 0$; $d_{j,i}^k \leftarrow 0$;
14: $\quad\quad$**end if**
15: $\quad\quad$**if** $T_{i,j}^k > T^k$ **or** $H_{i,j}^k > H^k$ **then**
16: $\quad\quad\quad d_{i,j}^k \leftarrow 0$;
17: $\quad\quad$**end if**
18: $\quad$**end for**
19: **end for**
20: remove the isolated MU from $\mathcal{N}^k$;
21: group $\mathcal{N}^k$ based on the connected components of $\mathbf{d}^k$ and the group set is denoted by G;
22: **return** $(G$, $\mathbf{d}^k)$;

we calculate the maximum bipartite matching by calling function **Allocation**$(\cdot)$ to obtain the allocation $\mathcal{M}^{k,h}$ of group g_h for resource k (Line 11).

As illustrated by Algorithm 4, function **Pricing**$(\cdot)$ follows the winner selection rule and payment determination rule of *McAfee Double Auction*. Let $p_0^k = (a_{t+1}^k + b_{t+1}^k)/2$. If $p_0^k \in [a_t^k, b_t^k]$, the first t sellers and the first t buyers win, and the charge/payment for each winning seller/buyer is p_0^k (Lines 4-9). Otherwise, the first $t-1$ sellers win, and the charge for each winning seller is a_t^k. Similarly, the first $t-1$ buyers win, and the payment for each winning buyer is b_t^k (Lines 11-16). Finally, **Pricing**$(\cdot)$ returns the winning seller set $E^{k,h,s}$, the winning buyer set $E^{k,h,b}$, and the payment profile $\mathbf{p}^k$ of resource k.

The allocation phase, which outputs the resource allocation through matching approach, is illustrated by Algorithm 5. We first divide each seller/buyer into multiple virtual sellers/buyers, and each is with resource supply/request of one unit (Lines 2-6). The sets of virtual sellers and virtual buyers are denoted by $VE^{k,h,s}$ and $VE^{k,h,b}$, respectively. Then we calculate the maximum matching $\mathcal{M}^{k,h}$ of the bigraph consisting of virtual sellers, virtual buyers, and the links in connectivity matrix $\mathbf{d}^k$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2021.3079258, IEEE Transactions on Vehicular Technology

Algorithm 3 : Transaction

Input: group set G, resource k, connectivity matrix $\mathbf{d}^k$, bid profile $\mathbf{B}^{N\times K}$, and ask profile $\mathbf{A}^{N\times K}$

foreach $g_h \in G$ **do**
 // Pricing Phase
 $\mathcal{N}^{k,h,s} \leftarrow \emptyset$; $\mathcal{N}^{k,h,b} \leftarrow \emptyset$;
 foreach $i \in g_h$ **do**
  if $s_i^k \neq 0$ **then** $\mathcal{N}^{k,h,s} \leftarrow \mathcal{N}^{k,h,s} \cup \{i\}$;
  else $\mathcal{N}^{k,h,b} \leftarrow \mathcal{N}^{k,h,b} \cup \{i\}$;
 end for
 sort all $i \in \mathcal{N}^{k,h,s}$ based on a_i^k in nondecreasing order;
 sort all $i \in \mathcal{N}^{k,h,b}$ based on b_i^k in nondecreasing order;
 Let t be the maximum index such that $a_i^k \leq b_i^k$ for all $i \in \mathcal{N}^{k,h,s}$ and $i \in \mathcal{N}^{k,h,b}$;
 $(E^{k,h,s}, E^{k,h,b}, \mathbf{p}^k)$ $\leftarrow$ **Pricing**$(t, k, \mathcal{N}^{k,h,s}, \mathcal{N}^{k,h,b}, \mathbf{B}^{N\times K}, \mathbf{A}^{N\times K})$;
 // Allocation Phase
 $\mathcal{M}^{k,h} \leftarrow$ **Allocation**$(E^{k,h,s}, E^{k,h,s}, \mathbf{B}^{N\times K}, \mathbf{A}^{N\times K}, \mathbf{d}^k)$;
 $\mathcal{M}^k \leftarrow \mathcal{M}^k \cup \mathcal{M}^{k,h}$;
end for
return $(\mathcal{M}^k$, $\mathbf{p}^k)$;

through function $\mathbf{MBM}(\cdot)$.

B. A Walk-Through Example

We still use the example in Fig. 3 to show how $TMTM$ works. Since grouping phase has been illustrated in Fig. 3, we only give the pricing phase and allocation phase for the left group in Fig. 3. The unit ask/bid price and the request/supply for each MU is given in Fig. 5.

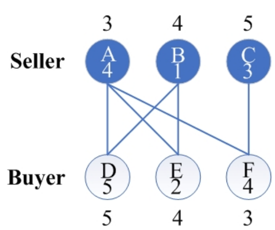


Fig. 5. Illustration for pricing phase and allocation phase, where the letters in the nodes represent the MUs, the number beside each node is the unit ask/bid price, the number in each node is the supply/request.

- Pricing: $t = 2$, $p_0^k = (a_C^k + b_F^k)/2 = 4 \in [a_B^k, b_E^k] = [4, 4]$, $E^{k,h,s} = \{A, B\}$, $E^{k,h,b} = \{D, E\}$, $p_A^k = p_B^k = p_D^k = p_E^k = p_0^k = 4$.
- Allocation: Calculate the maximum matching on the complete bipartite graph with $VE^{k,h,s} = \{A_1, A_2, A_3, A_4, B_1\}$ and $VE^{k,h,b} = \{D_1, D_2, D_3, D_4, D_5, E_1, E_2\}$. We have $|\mathcal{M}^k| = 5$.

Algorithm 4 : Pricing

Input: position t, seller set $\mathcal{N}^{k,h,s}$, buyer set $\mathcal{N}^{k,h,b}$, bid profile $\mathbf{B}^{N\times K}$, and ask profile $\mathbf{A}^{N\times K}$

$E^{k,h,s} \leftarrow \emptyset$; $E^{k,h,b} \leftarrow \emptyset$;
$p_0^k \leftarrow (a_{t+1}^k + b_{t+1}^k)/2$;
if $p_0^k \in [a_t^k, b_t^k]$ **then**
 foreach $i \in \mathcal{N}^{k,h,s}, i = 1, 2, ..., t$ **do**
  $p_i^k = p_0^k$; $E^{k,h,s} \leftarrow E^{k,h,s} \cup \{i\}$;
 end for
 foreach $i \in \mathcal{N}^{k,h,b}, i = 1, 2, ..., t$ **do**
  $p_i^k = p_0^k$; $E^{k,h,b} \leftarrow E^{k,h,b} \cup \{i\}$;
 end for
else
 foreach $i \in \mathcal{N}^{k,h,s}, i = 1, 2, ..., t-1$ **do**
  $p_i^k = a_t^k$; $E^{k,h,s} \leftarrow E^{k,h,s} \cup \{i\}$;
 end for
 foreach $i \in \mathcal{N}^{k,h,b}, i = 1, 2, ..., t-1$ **do**
  $p_i^k = b_t^k$; $E^{k,h,b} \leftarrow E^{k,h,b} \cup \{i\}$;
 end for
end if
return $(E^{k,h,s}$, $E^{k,h,b}$,$\mathbf{p}^k)$;

Algorithm 5 : Allocation

Input: winning seller set $E^{k,h,s}$, winning buyer set $E^{k,h,b}$, bid profile $\mathbf{B}^{N\times K}$, and ask profile $\mathbf{A}^{N\times K}$, connectivity matrix $\mathbf{d}^k$

$VE^{k,h,s} \leftarrow \emptyset$; $VE^{k,h,b} \leftarrow \emptyset$;
foreach $i \in E^{k,h,s}$ **do**
 divide i into s_i^k virtual sellers, and put them into set $VE^{k,h,s}$;
end for
foreach $i \in E^{k,h,b}$ **do**
 divide i into r_i^k virtual sellers, and put them into set $VE^{k,h,b}$;
end for
$\mathcal{M}^{k,h} \leftarrow$ **MBM**$(VE^{k,h,s}, VE^{k,h,b}, \mathbf{d}^k)$;
return $\mathcal{M}^{k,h}$;

We randomly choose one of the optimal solutions of maximum matching, e.g., $Q_{D,A}^k = 4$, $Q_{D,B}^k = 1$ with price $p_A^k = p_B^k = p_D^k = 4$ for per unit of resource k.

C. Mechanism Analysis

In the following, we present the theoretical analysis, demonstrating that $TMTM$ can achieve the desirable properties.

Lemma 1. *$TMTM$ is computationally efficient.*

Proof: It suffices to show that both Algorithm 2 and Algorithm 3 are computationally efficient.

In Algorithm 2, the for-loop (Lines 2-9) takes $O(N^2)$ time. The for-loop (Lines 10-19) also takes $O(N^2)$ time.

Computing the connected components (Line 21) takes $O(N^2)$ time. Thus the running time of Algorithm 2 is $O(N^2)$.

The running time of Algorithm 3 is dominated by function **Allocation**$(\cdot)$ (Lines 11), and the running time of function **Allocation**$(\cdot)$ is dominated by computing the maximum matching for the bigraph (Line 7 of Algorithm 5). Computing the maximum matching usually takes $O(\varphi N \sum_{i,j\in\mathcal{N}} d_{i,j}^k \varphi)$ time, where φN is the total number of virtual sellers/buyers, $\sum_{i,j\in\mathcal{N}} d_{i,j}^k \varphi$ is the number of links between virtual sellers and virtual buyers, φ is the maximum supply/request of MU. Algorithm 3 executes function **Allocation**$(\cdot)$ for each group iteratively, thus the running time of Algorithm 3 is $O(\varphi^2 N^2 \sum_{i,j\in\mathcal{N}} d_{i,j}^k)$.

Since *TMTM* executes Algorithm 2 and Algorithm 3 for each resource, the computation complexity of *TMTM* is $O(K\varphi^2 N^2 \sum_{i,j\in\mathcal{N}} d_{i,j}^k)$. ■

Lemma 2. *TMTM is individually rational.*

Proof: Based on equation (8) and equation (9), we only need to prove $p_i^k - c_i^k \geq 0$ and $v_i^k - p_i^k \geq 0$ for any winning seller $i \in E^{k,h,s}$ and any winning buyer $i \in E^{k,h,b}$, respectively.

For any winning seller $i \in E^{k,h,s}$, we discuss the following two cases:

Case 1 (Lines 4-6 of Algorithm 4): $i \in \{1, 2, ..., t\}$, we have $p_i^k = p_0^k \geq a_t^k \geq a_i^k$.

Case 2 (Lines 11-13 of Algorithm 4): $i \in \{1, 2, ..., t-1\}$, we have $p_i^k = a_t^k \geq a_i^k$.

For any winning buyer $i \in E^{k,h,b}$, we discuss the following two cases:

Case 1 (Lines 7-9 of Algorithm 4): $i \in \{1, 2, ..., t\}$, we have $p_i^k = p_0^k \leq b_f^k \leq b_i^k$.

Case 2 (Lines 14-16 of Algorithm 4): $i \in \{1, 2, ..., t-1\}$, we have $p_i^k = b_f^k \leq b_i^k$.

Since *TMTM* is value/cost generally truthful (will be proved in Lemma 4), we have $a_i^k = c_i^k$ and $b_i^k = v_i^k$. This is sufficient to guarantee $u_i \geq 0$ for any $i \in \mathcal{N}$. ■

Lemma 3. *TMTM is budget balanced.*

Proof: For any winning buyer $i \in E^{k,h,b}$ and winning seller $j \in E^{k,h,s}$, we discuss the following two cases:

Case 1 (Lines 4-9 of Algorithm 4): $i, j \in \{1, 2, ..., t\}$, we have $p_i^k = p_j^k = p_0^k$.

Case 2 (Lines 11-16 of Algorithm 4): $i, j \in \{1, 2, ..., t-1\}$, we have $p_i^k = b_t^k$, $p_j^k = a_t^k$. Since $a_t^k \leq b_t^k$, we have $p_i^k \geq p_j^k$.

This is sufficient to guarantee $u_0 = \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{N}, r_i^k>0} \sum_{j\in\mathcal{N}, s_j^k>0} (p_i^k - p_j^k) Q_{i,j}^k \geq 0$. ■

Lemma 4. *TMTM is value/cost generally truthful.*

Proof: We consider the cost general truthfulness of any seller $i \in \mathcal{N}^{k,h,s}$ (similar for value general truthfulness of any buyer $i \in \mathcal{N}^{k,h,b}$). Since the allocation phase of *TMTM* outputs one of the optimal solutions of maximum matching randomly, the allocation phase is independent of the unit ask price, i.e., any seller $i \in \mathcal{N}^{k,h,s}$ of double auction cannot increase the expectation of $\sum_{j\in\mathcal{N}} Q_{j,i}^k$ for any resource $k \in \mathcal{K}$ in the allocation phase. Since $u_i^s = \sum_{k\in\mathcal{K}, s_i^k>0} ((p_i^k - c_i^k) \sum_{j\in\mathcal{N}} Q_{j,i}^k)$, we only need to prove that reporting false cost cannot increase the value of $(p_i^k - c_i^k)$ for any seller $i \in \mathcal{N}^{k,h,s}$ and any resource $k \in \mathcal{K}$ in the pricing phase.

If $i \in E^{k,h,s}$, changing the unit ask price cannot change anything since the payment is independent of its unit ask price. This means the value of $(p_i^k - c_i^k)$ is also unchanged.

If $i \notin E^{k,h,s}$, there are two cases:

Case 1: $i = t$. This means $a_i^k \geq a_{t-1}^k$ and $p_0^k \notin [a_t^k, b_t^k]$. i must ask $\widetilde{a_i^k} < a_{t-1}^k$ to win. We have two cases further:

(1) $p_0^k > b_t^k$, there must be $p_0^k \notin [a_{t-1}^k, b_t^k]$. We have:

$$p_i^k - a_i^k = a_{t-1}^k - a_i^k \leq a_i^k - a_i^k = 0 \tag{12}$$

(2) $p_0^k < a_t^k$, we have

$$p_i^k - a_i^k = \begin{cases} a_{t-1}^k - a_i^k \leq a_i^k - a_i^k = 0, & \text{if } p_0^k \in [a_{t-1}^k, b_t^k] \\ p_0^k - a_i^k < a_t^k - a_i^k = 0, & \text{otherwise} \end{cases} \tag{13}$$

Case 2: $i > t$. This means $a_i^k \geq a_t^k$ and $p_0^k \in [a_t^k, b_t^k]$. i must ask $\widetilde{a_i^k} < a_t^k$ to win. We have two cases further:

(1) $\frac{a_t^k + b_{t+1}^k}{2} \in [a_{t-1}^k, b_t^k]$, we have:

$$p_i^k - a_i^k = \frac{a_t^k + b_{t+1}^k}{2} - a_i^k < a_t^k - a_i^k \leq 0 \tag{14}$$

(2) $\frac{a_t^k + b_{t+1}^k}{2} \notin [a_{t-1}^k, b_t^k]$, we have: $p_i^k - a_i^k = a_{t-1}^k - a_i^k \leq 0$ if $\widetilde{a_i^k} < a_{t-1}^k$, i lose the auction otherwise.

In conclusion, no seller can increase the value of $(p_i^k - c_i^k)$ by submitting a false cost in the pricing phase. ■

Lemma 5. *TMTM is request/supply truthful.*

Proof: We consider the supply truthfulness of any seller $i \in \mathcal{N}^{k,h,s}$ (similar for request truthfulness of any buyer $i \in \mathcal{N}^{k,h,b}$).

Since the pricing phase is independent of the supply, any seller $i \in \mathcal{N}^{k,h,s}$ cannot increase the value of $(p_i^k - c_i^k)$ by reporting false supply. Since $u_i^s = \sum_{k\in\mathcal{K}, s_i^k>0} ((p_i^k - c_i^k) \sum_{j\in\mathcal{N}} Q_{j,i}^k)$, we only need to prove that reporting false supply cannot increase the value of $\sum_{j\in\mathcal{N}} Q_{j,i}^k$ for any resource $k \in \mathcal{K}$ in the allocation phase.

We assume that seller i submits supply $s_i^k \neq \widetilde{s_i^k}$. There are two cases:

Case 1: $\sum_{j\in\mathcal{N}} Q_{j,i}^k = \widetilde{s_i^k}$, i.e., all supply of seller i have been allocated. In this case, seller i obtains the highest value of $\sum_{j\in\mathcal{N}} Q_{j,i}^k$ (Note that the utility of seller i will be zero if $\sum_{j\in\mathcal{N}} Q_{j,i}^k > \widetilde{s_i^k}$).

Case 2: $\sum_{j\in\mathcal{N}} Q_{j,i}^k < \widetilde{s_i^k}$, i.e., some resource of seller i have not been allocated. In this case, all requests of buyers with links to seller i must have been satisfied. If not, we can add a transaction between i and the unsatisfied buyers, thus obtaining a larger matching. This results in contradiction. If seller i submits $s_i^k > \widetilde{s_i^k}$, nothing changes in the allocation phase because all requests of buyers with link to seller i have been satisfied. If seller i submits $s_i^k < \widetilde{s_i^k}$, the value of $\sum_{j\in\mathcal{N}} Q_{j,i}^k$ does not change or decreases.

In conclusion, no seller can increase the value of $\sum_{j\in\mathcal{N}} Q_{j,i}^k$ by submitting a false request/supply in the allocation phase. ■

The above five lemmas together prove the following theorem.

Theorem 1. *TMTM is computationally efficient, individually rational, budget balanced, value/cost generally truthful, and request/supply truthful.*

VI. Performance Evaluation

We have conducted simulations to investigate the performance of Honest Model and *TMTM* for strategic MUs on the real experience data.

A. Simulation Setup

The simulations are based on SIGCOMM 2009 trace [25], which contains traces of 76 participants through *MobiClique* application at SIGCOMM 2009 conference in Barcelona, Spain from August 17th to August 21st, 2009. In this data set, each device performs a periodic Bluetooth device discovery every 120 seconds. Upon discovering new contacts, the devices form a *RFCOMM* link on a preconfigured channel for data communications. The total number of time slots is 320438. We divide the whole data set into 17 time periods. The length of each time period is 20000. We extract the contacts in each time period, and consider the two devices are in proximity if they have contacts in this time period. We classify these time periods into *Weak Connectivity*, *Medium Connectivity* and *Strong Connectivity* based on the contracts of the participants.

- *Weak Connectivity*: Time periods with [0, 100] effective contact logs.
- *Medium Connectivity*: Time periods with [101, 800] effective contact logs.
- *Strong Connectivity*: Time periods with more than 800 effective contact logs.

As a result, there are 7 *Weak Connectivity* time periods, 5 *Medium Connectivity* time periods and 5 *Strong Connectivity* time periods. Since the impact of the connectivity is large, we use the *Strong Connectivity* time periods as the default setting. Each measurement is averaged over all 5 *Strong Connectivity* time periods. However, we will vary the connectivity to explore the impacts on designed algorithms. Fig. 6 depicts the topologies of all 76 participants for the realizations of *Weak Connectivity*, *Medium Connectivity* and *Strong Connectivity*, respectively.

TABLE II
Parameter Settings

Parameter	Value
r_i^k, s_i^k	[1, 10]
D_i^k	[50, 200] KB
T^1, T^2, T^3	3 s, 1.5 s, 1.3 s
H^1, H^2, H^3	4.1×10^{-5} Wh, 4×10^{-5} Wh, 4.4×10^{-5} Wh
f	0.1GHz
$W_{i,j}$	10MHz
$l_{i,j}$	[3, 10] m
Pt_i	100 mW
$W_{j,0}$	15 MHz
α	10^{-28}
β	11dB
σ^2	10^{-10} mW
Market Activity	100%
Supply-request Ratio	1:1

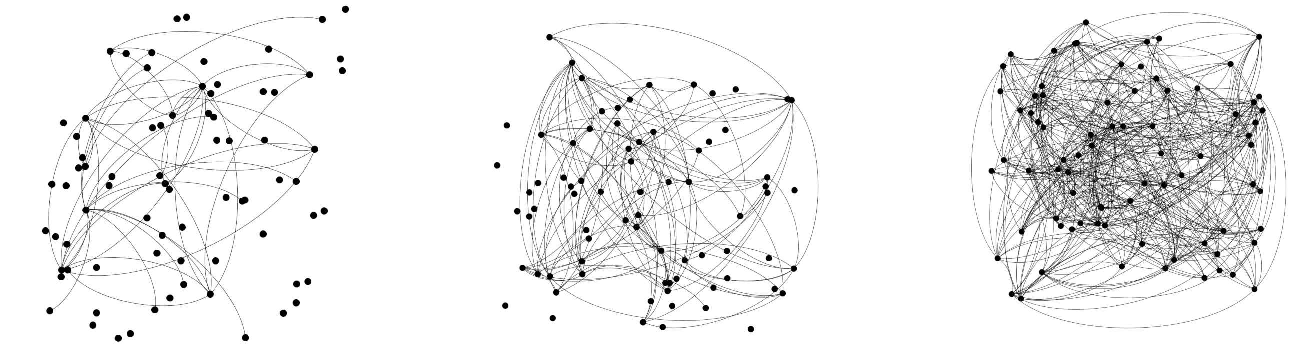
The cost/value of each user is selected randomly from the auction dataset [26], which contains 5017 bid prices for Palm Pilot M515 PDA from eBay. We consider there are 3 types of resources: computation resource, communication resource and storage resource in the task offloading system. We define *Market Activity* and *Supply-request Ratio* to simulate the different markets:

- *Market Activity*: The probability of each MU to supply or request resource. Market Activity represents the willingness of MUs to participate in the task offloading system.
- *Supply-request Ratio*: The ratio of the number of buyers to the number of sellers for any given resource.

The default settings of parameters are given in TABLE II. All the simulations were run on a Centos 7 machine with Intel(R) Xeon(R) CPU E5-2630 2.6GHz and 128 GB memory. Each measurement is averaged over 100 instances.

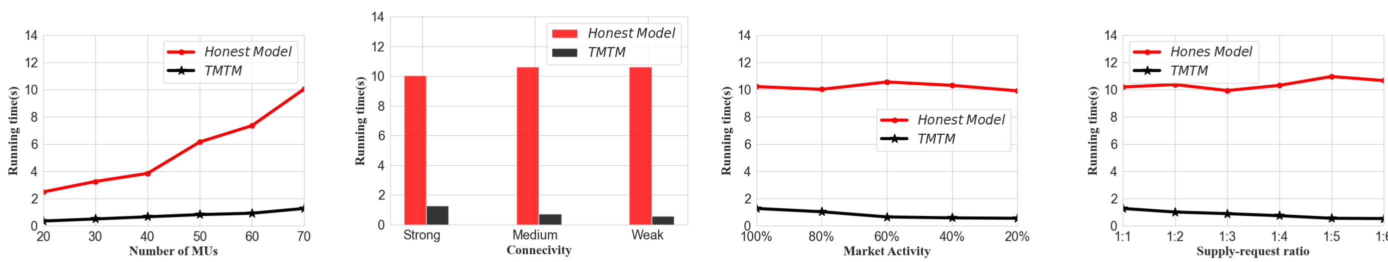
B. Running Time

As shown in Fig. 7, *TMTM* only takes 13% running time of Honest Model averagely. *TMTM* can be terminated within 1.1 seconds in all our simulations, exhibiting prominent superiority in terms of computational efficiency. We also can see that the running time of Honest Model is insensitive to the connectivity, *Market Activity*, and *Supply-request Ratio*. However, *TMTM* takes less time when the connectivity becomes weak. This is because the grouping



(a) (b) (c)

Fig. 6. Network Realizations with different Connectivities. (a) Realization of *Weak Connectivity* with 43 links in time period [60000, 80000]. (b) Realization of *Medium Connectivity* with 475 links in time period [140000, 160000]. (c) Realization of *Strong Connectivity* with 998 links in time period [100000, 120000].



(a) (b) (c) (d)

Fig. 7. Running time. (a) Running time versus MUs. (b) Running time versus connectivity. (c) Running time *Market Activity*. (d) Running time versus *Supply-request Ratio*.

phase removes the isolated MUs from the MU set and essentially reduces the number of MUs in the system. The running time of *TMTM* also decreases with the decreasing *Market Activity* since the number of MUs of the system decreases with decreasing *Market Activity*. Fig. 7(d) shows that *TMTM* takes less time when the *Supply-request ratio* becomes unbalanced. The unbalance of buyers and sellers will reduce the number of winners of double auction. This means that fewer MUs can enter the allocation phase. Since the time complexity of *TMTM* is dominated by the allocation phase according to Lemma 1, *TMTM* will take less time.

C. Number of Transactions

We measure the performance of Honest Model and *TMTM*, and compare it with following three benchmark mechanisms:

- *Non Grouping*: There is no grouping phase in this mechanism. The pricing phase and allocation phase are the same as those of *TMTM*.
- *Single-unit McAfee*: Divide each seller/buyer into multiple virtual sellers/buyers, and each with resource request/supply of single unit for each resource $k \in \mathcal{K}$. Then apply *McAfee Double Auction* to select winners and determine the price.
- *Coalition* [4]: This mechanism enables one-to-one communication and computation resources sharing in an OFDMA cellular network, and uses coalition game [27] to find the reciprocal resource sharing peers.

Fig. 8 compares the number of the transactions achieved by Honest Model, *TMTM*, *Non Grouping* and *Single-unit McAfee*. Recall that we use the term “allocation” in the honest model since there is no money transfer among MUs. For convenience, we use the term “transaction” uniformly in Fig. 8. In all cases, Honest Model can obtain the most transactions, which is 189% of *TMTM* averagely since Honest Model outputs the optimal solution for transaction maximization problem. However, *TMTM* achieves the important economic property of truthfulness of strategic MUs and takes much less time than Honest Model. Comparing with *Non Grouping* and *Single-unit McAfee*, *TMTM* outputs the most transactions (with average improvement 131% and 33%, respectively). *Single-unit McAfee* divides the seller/buyer into multiple virtual sellers/buyers of sin-

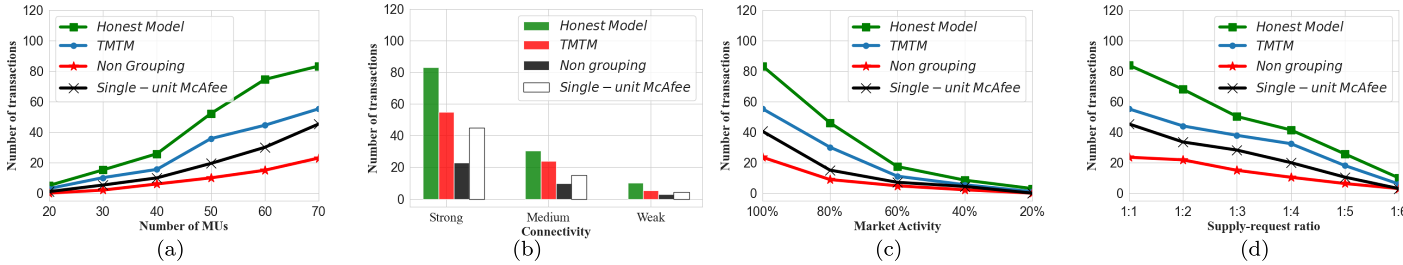


Fig. 8. Number of transactions with 3 types of resources. (a) Number of transactions versus number of MUs. (b) Number of transactions versus connectivity. (c) Number of transactions versus *Market Activity*. (d) Number of transactions versus *Supply-request Ratio*.

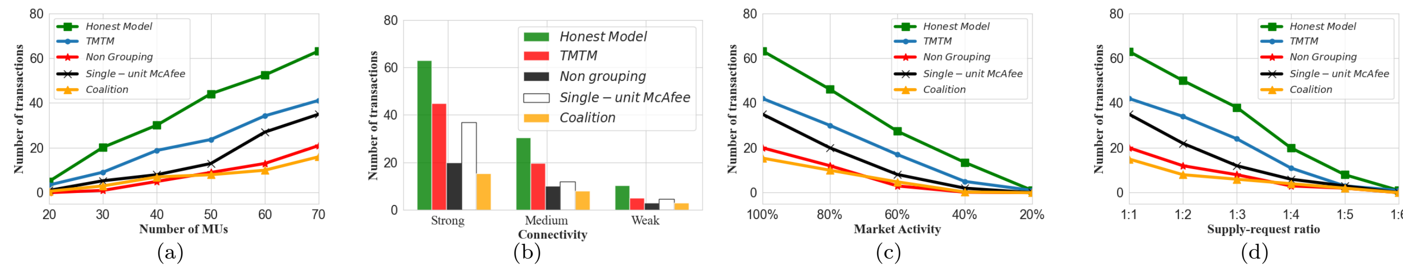


Fig. 9. Number of transactions with 2 types of resources. (a) Number of transactions versus number of MUs. (b) Number of transactions versus connectivity. (c) Number of transactions versus *Market Activity*. (d) Number of transactions versus *Supply-request Ratio*.

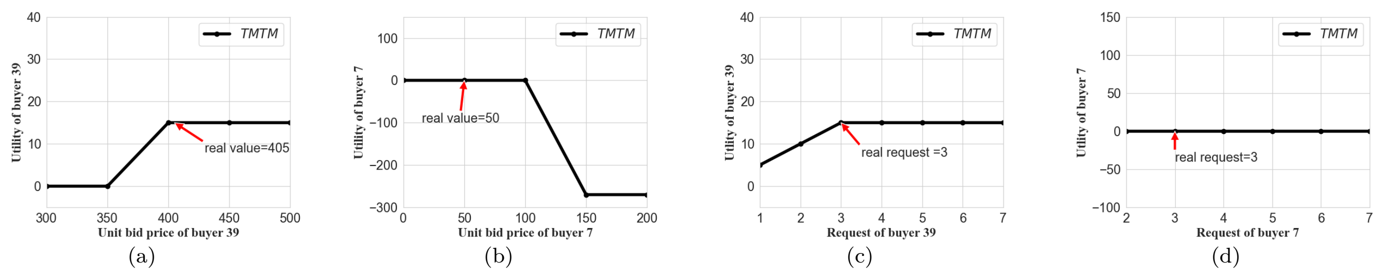


Fig. 10. Truthfulness of transactions of computation resource. (a) Utility of winning buyer 39 with value 405. (b) Utility of losing buyer 7 with value 50. (c) Utility of winning buyer 39 with real request 3. (d) Utility of losing buyer 7 with real request 3.

gle unit of resource, and tries to translate the multi-unit double auction problem to the single-unit double auction problem. However, it may result in reduction of winners if the most resources are requested or supplied by the minority of MUs. More important, the failure of guaranteeing request/supply truthfulness makes *Single-unit McAfee* less attractive. *Non Grouping* mechanism does not conduct the preprocessing of MUs before pricing phase. Thus the invalid MUs, including the MUs from different connected components and the isolated MUs, will decrease the efficiency of both double auction and matching. As shown in Fig. 8, the number of transactions of *Non Grouping* is much fewer than those of *TMTM* and *Single-unit McAfee*. This means that the impact of invalid MUs on the transactions is great, and the grouping phase is an indispensable module of multi-resource transaction mechanism.

Because the comparison algorithm is only applicable for 2 types of resources, we measure the number of the transactions achieved by *Honest Model*, *TMTM*, *Non Grouping*, *Single-unit McAfee* and *Coalition* with 2 types of resources: computation resource and communication resource. As shown in Fig. 9, *TMTM* can obtain more transactions than *Coalition* in all cases (297% of the *Coalition* averagely) because one task can only be offloaded to one MU in *Coalition*. While in other algorithms, one task can be offloaded to multiple MUs, and can take full advantage of surplus resources. In addition, *TMTM* follows auction framework and satisfies important economic properties, which make it more practical in real resource transaction markets.

D. Truthfulness

Due to the space constraints, we only verify the truthfulness of buyers in transactions of computation resource ($k = 1$), and illustrate the results in Fig. 10. We verified the truthfulness of *TMTM* by randomly picking two buyers and allowing them to submit false unit bid price and resource request. We can see that the winning buyer 39 always obtain its maximum utility of 15 if it bids the real value $v_{39}^1 = 405$ and real request $\widetilde{r_{39}^1} = 3$. Accordingly, the losing buyer 7 always obtains nonnegative utility if it bids the real value $v_7^1 = 50$ and real request $\widetilde{r_7^1} = 3$. For the case of Fig. 10(c), the number of allocated resources is 3. According to Lemma 5, no resource can be sold in the system. Thus if buyer 39 submits $r_{39}^1 > 3$, nothing changes in the allocation phase because all resources of sellers who can link to buyer 39 have been sold.

VII. Discussion

In this section, we discuss the implementation problems of *TMTM*.

The allocation model is also applicable to the multi-cell scenario. The MUs can submit the resource request/supply to its BS. If any MU is covered by multiple BSs, it can participate in the resource allocation in any BS or submit the divided resource request/supply to multiple BSs.

In reality, when the BS launches the resource transaction service, the resource request and resource supply are collected. Then the BS performs *TMTM* based on the newly collected information and the current network topology. The resource transaction algorithm can be performed periodically on-demand. The periodic execution makes the algorithms be adaptable to the time-varying environments.

However, the execution of both algorithms and offloaded tasks need time. For the fast-varying wireless networks, the network topology may be changed before obtaining the results of offloaded tasks. This is a common and interesting issue for many edge computing paradigms for mobile users. We discuss some possible solutions here:

(1) To reduce processing time of tasks, the tasks can be divided into small tasks, which can be processed fast. However, not all tasks are dividable.

(2) We can remove the fast-varying wireless connections, which cannot maintain the task offloading, in the grouping phase. First, the running time of *TMTM* can be estimated. Then the connections should be stable for some time until the tasks are finished. For any resource , the task execution time can be estimated based on the latency constraint T^k, e.g., several times of T^k. Moreover, the trajectory prediction methods [28, 29] for MUs can be adopted to estimate the stability of links. If the connections cannot keep stable for the time of algorithm execution and task execution, we can remove these unstable wireless connections directly in the grouping phase.

(3) While noticing the upcoming disconnection, the seller can stop task processing and return the current results to the buyer, and obtain the payment according to the actually provided resources. Accordingly, after receiving the partial results of this task, the buyer can submit the new resource request in the next auction. Notice that *TMTM* is truthful, and the unit resource payment only depends on the cost/value of resource, rather than the number of resources actually provided. So, *TMTM* is still individually rational, budget balanced and truthful. However, this method is invalid if the partial results of tasks are valueless, e.g., the indivisible tasks. In this case, some of the tasks cannot be completed though *TMTM* still works and is individually rational, budget balanced and truthful.

VIII. Conclusion

In this paper, we have studied the multi-resource transaction mechanisms for P2P Task Offloading in honest MU model and strategic user model, respectively. We have formulated and solved the allocation maximization problem with latency and energy consumption constraints for the honest mobile users. We have shown that the solution for honest users can output 189% resource transactions of that for the strategic users. We have modeled a double auction based P2P task offloading system and have formulated the transaction maximization problem for the strategic mobile users. We have designed a truthful multi-resource transaction mechanism, *TMTM*, integrating with *McAfee Double Auction* and matching approach for the strategic mobile users. Through both rigorous theoretical analysis and extensive simulations, we have demonstrated that *TMTM* satisfies the desirable properties of computational efficiency, individual rationality, budget balance, value/cost general truthfulness, and request/supply truthfulness. Moreover, *TMTM* shows prominent advantage in terms of the number of resource transactions, comparing with *Single-unit McAfee*, the solution without considering the connectivity, and the coalition game-based mechanism.

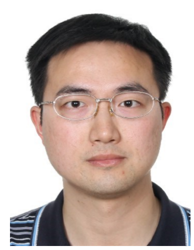
References

[1] Y. Mao, C. You, J. Zhang, K. Huang and K. B. Letaief, "A Survey on Mobile Edge Computing: The Communication Perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322-2358, 2017.

[2] T. Wang, Y. Sun, L. Song and Z. Han, "Social Data Offloading in D2D-Enhanced Cellular Networks by Network Formation Games," *IEEE Transactions on Wireless Communications*, vol. 14, no. 12, pp. 7004-7015, 2015.

[3] M. Yang, S. Lim, H. Park and N. Park, "Solving the data overload: Device-to-device bearer control architecture for cellular data offloading," *IEEE Vehicular Technology Magazine*, vol. 8, no. 1, pp. 31-39, 2013.

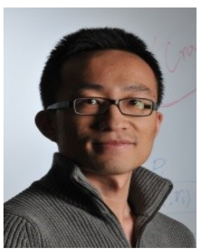
[4] Y. Cao, C. Long, T. Jiang and S. Mao, "Share communication and computation resources on mobile devices: a social awareness perspective," *IEEE Wireless Communications*, vol. 23, no. 4, pp. 52-59, 2016.
[5] C. Long, Y. Cao, T. Jiang and Q. Zhang, "Edge computing framework for cooperative video processing in multimedia IoT systems," *IEEE Transactions on Multimedia*, vol. 20, no. 5, pp. 1126-1139, 2017.
[6] W. Chen, D. Wang and K. Li, "Multi-User Multi-Task Computation Offloading in Green Mobile Edge Cloud Computing," *IEEE Transactions on Services Computing*, vol. 12, no. 5, pp. 726-738, 2019.
[7] Y. Wu, J. Chen, L. P. Qian, J. Huang and X. S. Shen, "Energy-Aware Cooperative Traffic Offloading via Device-to-Device Cooperations: An Analytical Approach," *IEEE Transactions on Mobile Computing*, vol. 16, no. 1, pp. 97-114, 2017.
[8] Y. Pan, C. Pan, H. Zhu, Q. Z. Ahmed, M. Chen and J. Wang, "On Consideration of Content Preference and Sharing Willingness in D2D Assisted Offloading," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 4, pp. 978-993, 2017.
[9] Q. Zhang, L. Gui, F. Hou, J. Chen, S. Zhu and F. Tian, "Dynamic Task Offloading and Resource Allocation for Mobile-Edge Computing in Dense Cloud RAN," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3282-3299, 2020.
[10] Y. Sun, C. Song, S. Yu, Y. Liu, H. Pan and P. Zeng, "Energy-Efficient Task Offloading Based on Differential Evolution in Edge Computing System With Energy Harvesting," *IEEE Access*, vol. 9, pp. 16383-16391, 2021.
[11] T. X. Tran and D. Pompili, "Joint Task Offloading and Resource Allocation for Multi-Server Mobile-Edge Computing Networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 856-868, 2019.
[12] Z. Zheng, Y. Gui, F. Wu and G. Chen, "STAR: Strategy-Proof Double Auctions for Multi-Cloud, Multi-Tenant Bandwidth Reservation," *IEEE Transactions on Computers*, vol. 64, no. 7, pp. 2071-2083, 2015.
[13] W. Xu, H. Huang, Y. Sun, F. Li, Y. Zhu and Shukui, "DATA: A double auction based task assignment mechanism in crowdsourcing systems," in *Proc. of EAI International Conference on Communications and Networking in China*, 2013, pp. 172-177.
[14] X. Peng, K. Ota and M. Dong, "Multiattribute-Based Double Auction Toward Resource Allocation in Vehicular Fog Computing," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 3094-3103, 2020.
[15] W. Sun, J. Liu, Y. Yue and H. Zhang, "Double Auction-Based Resource Allocation for Mobile Edge Computing in Industrial Internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4692-4701, 2018.
[16] G. Gao, M. Xiao, J. Wu, H. Huang, S. Wang and G. Chen, "Auction-based VM Allocation for Deadline-Sensitive Tasks in Distributed Edge Cloud," *IEEE Transactions on Services Computing*, doi: 10.1109/TSC.2019.2902549, 2019
[17] R. McAfee, "A dominant strategy double auction," *Journal of economic Theory*, vol. 56, no. 2, pp. 434-450, 1992.
[18] Jan. 2020. [Online]. https://www.ibm.com/analytics/cplex-optimizer.
[19] X. Zhou and H. Zheng, "TRUST: A General Framework for Truthful Double Spectrum Auctions," in *Proc. of IEEE INFOCOM*, 2009, pp. 999-1007.
[20] J. Xu, J. Xiang and D. Yang. "Incentive Mechanisms for Time Window Dependent Tasks in Mobile Crowdsensing," *IEEE Transactions on Wireless Communications*, vol. 14, no.11, pp. 6353-6364, 2015.
[21] J. Xu, Z. Rao, L. Xu, D. Yang and T. Li. "Incentive Mechanism for Multiple Cooperative Tasks with Compatible Users in Mobile Crowd Sensing via Online Communities," *IEEE Transactions on Mobile Computing*, vol. 19, no. 7, pp. 1618-1633, 2020.
[22] L. Jiang, X. Niu, J. Xu, D. Yang and L. Xu, "Incentive Mechanism Design for Truth Discovery in Crowdsourcing with Copiers," *IEEE Transactions on Services Computing*, doi:10.1109/TSC.2021.3075741, 2021.
[23] W. Douglas, "Introduction to Graph Theory (2nd ed.)," Prentice Hall, Chapter 3, 1999.
[24] L. Ford and D. Fulkerson, "Maximal flow through a network," *Canadian Journal of Mathematics*, vol. 8, no. 3, pp. 399-404, 1956.
[25] A. Pietilainen and C. Diot, CRAWDAD dataset thlab/sigcomm2009 (v. 2012-07-15), [Online]. https://crawdad.org/thlab/sigcomm2009/20120715, Jul 2012.
[26] Jan. 2020. [Online]. https://www.modelingonlineauctions.com/datasets.
[27] Y. Chen, X. Gong, R. Ou, L. Duan and Q. Zhang, "Crowdcaching: Incentivizing D2D-enabled caching via coalitional game for IoT," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 5599-5612, 2020.
[28] K. Zhang, Y. Mao, S. Leng, Y. He and Y. Zhang. "Mobile-edge computing for vehicular networks: A promising network paradigm with predictive off-loading." *IEEE Vehicular Technology Magazine*, vol. 12, no. 2, pp. 36-44, 2017.
[29] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, F. Li and S. Savarese. "Social lstm: Human trajectory prediction in crowded spaces." In *Proc. of IEEE conference on computer vision and pattern recognition*, 2016, pp. 961-971.



Weifeng Lu was born in 1979. He received the Ph.D. degree in Communication and Information System from Nanjing University of Posts and Telecommunications University in 2007. Since 2012, he has been working as a Postdoctoral Researcher in Information Science and Engineering of Southeast University. His research interest is in radio resource management in device-to-device networks and edge computing, etc. He is currently an associate professor in the School of Computer Science at Nanjing University of Posts and Telecommunications.

Shitao Zhang received the B.E. degree in the Department of Computer Science and Technology from Hebei Normal University, Shijiazhuang, China, in 2016. She is currently pursuing the M.E. degree in computer technology from Nanjing University of Posts and Telecommunications, Nanjing, China. Her main research interests is in edge computing.

Jia Xu (M'15) received the M.S. degree in School of Information and Engineering from Yangzhou University, Jiangsu, China, in 2006 and the PhD. Degree in School of Computer Science and Engineering from Nanjing University of Science and Technology, Jiangsu, China, in 2010. He is currently a professor in the School of Computer Science at Nanjing University of Posts and Telecommunications. He was a visiting Scholar in the Department of Electrical Engineering & Computer Science at Colorado School of Mines from Nov. 2014 to May. 2015. His main research interests include crowdsourcing, edge computing and wireless sensor networks. Prof. Xu has served as the PC Co-Chair of SciSec 2019, Organizing Chair of ISKE 2017, TPC member of Globecom, ICC, MASS, ICNC, EDGE. He currently serves as the Publicity Co-Chair of SciSec 2021.



Dejun Yang (M'13 - SM'19) received the B.S. degree in computer science from Peking University, Beijing, China, in 2007 and the Ph.D. degree in computer science from Arizona State University, Tempe, AZ, USA, in 2013. Currently, he is an Associate Professor of computer science with Colorado School of Mines, Golden, CO, USA. His research interests include Internet of things, networking, and mobile sensing and computing with a focus on the application of game theory, optimization, algorithm design, and machine learning to resource allocation, security and privacy problems. Prof. Yang has served as the TPC Vice-Chair for Information Systems for IEEE International Conference on Computer Communications (INFOCOM) and currently serves an Associate Editor for the IEEE Internet of Things Journal (IoT-J). He has received the IEEE Communications Society William R. Bennett Prize in 2019 (best paper award for IEEE/ACM Transactions on Networking (TON) and IEEE Transactions on Network and Service Management in the previous three years), Best Paper Awards at IEEE Global Communications Conference (GLOBECOM) (2015), IEEE International Conference on Mobile Ad hoc and Sensor Systems (2011), and IEEE International Conference on Communications (ICC) (2011 and 2012), as well as a Best Paper Award Runner-up at IEEE International Conference on Network Protocols (ICNP) (2010).

Lijie Xu received his Ph.D. degree in the Department of Computer Science and Technology from Nanjing University, Nanjing, in 2014. He was a research assistant in the Department of Computing at the Hong Kong Polytechnic University, Hong Kong, from 2011 to 2012. He is currently an associate professor in the School of Computer Science at Nanjing University of Posts and Telecommunications, Nanjing. His research interests are mainly in the areas of wireless sensor networks, ad-hoc networks, mobile and distributed computing, and graph theory algorithms.