Edgeconomics: Price Competition and Selfish Computation Offloading in Multi-Server Edge Computing Networks

Ziya Chen, Qian Ma, Lin Gao, and Xu Chen

***Abstract*—As edge computing provides crucial support for delay-sensitive and computation-intensive applications, many business entities deploy their own edge servers to compete for users, which forms multi-server edge computing networks. However, no prior work studies the competition among heterogeneous edge servers and how the competition affects users' selfish computation offloading behaviors in such a network from an economic perspective. In this paper, we model the interactions between edge servers and users as a two-stage game. In Stage I, edge servers with heterogeneous marginal costs set their service prices to compete for users, and in Stage II, each user selfishly offloads its task to one of the edge servers or the remote cloud. Analyzing the equilibrium of the two-stage game is challenging due to edge servers' heterogeneity and the congestion effect caused by resource sharing among users. We first prove that in Stage II, users' selfish computation offloading game is a potential game and admits a unique Nash equilibrium (NE), for which we derive the explicit expression. We then analyze edge servers' price competition game in Stage I and characterize the conditions for the uniqueness of the NE. We show that at equilibrium, users only choose low-priced edge servers, and hence edge servers with low marginal costs can win the price competition, which reflects the improvement of economic efficiency in competitive markets. Moreover, it is surprising that the equilibrium prices do not monotonically increase with the task execution delay. This is because a long execution delay gives a chance to edge servers with high marginal costs to win the competition, which results in more fierce competition among edge servers.**

I. Introduction

With the development of mobile devices and Internet of Things (IoT) technologies, various applications (e.g., autonomous driving, interactive gaming with virtual reality, and face recognition) emerge and gain more and more popularity among mobile users [1]. These applications are usually delay-sensitive and computation-intensive. Cloud computing is a traditional approach to execute these application tasks by using ample computing resources on the cloud. However, transmitting the input data of the application tasks to the remote cloud over the backbone networks may lead to unbearable delays to users [2]. Edge computing appears as a promising solution to this dilemma. It performs data processing on edge servers which are close to users, and hence avoids the transmission delay at the backhaul of the network [3].

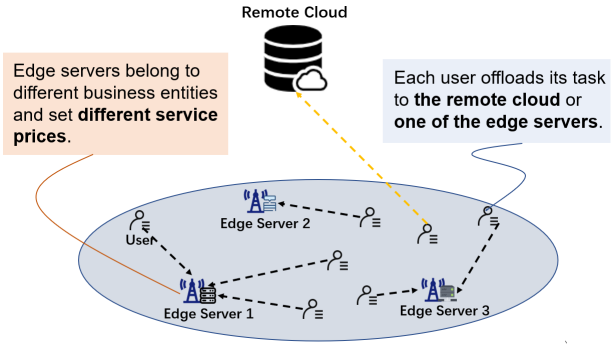

Fig. 1. An example of the multi-server edge computing network

As an important technology to improve users' experiences of the delay-sensitive and computation-intensive applications, edge computing becomes the focus of many companies. Major telecom operators actively launch their edge computing platforms, such as 5G Edge by Verizon and OpenSigma by China Mobile. Tech giants also provide many commercial products and services of edge computing, such as Azure Edge Zones from Microsoft, EdgeGallery by HUAWEI, and Wavelength framework by Amazon. According to IDC, a quarter of organizations will integrate edge computing with applications built on cloud platforms to improve business agility by 2024 [4]. As a result, in some areas with massive mobile users, there may be multiple edge servers deployed by different business entites [5] [6]. We show an example of the multi-server edge computing network with a remote cloud and multiple edge servers in Fig. 1. Despite the commercial success of edge computing in practice, little work performs comprehensive economic analysis for multi-server edge computing networks.

In a multi-server edge computing network, mobile users who generate delay-sensitive and computation-intensive tasks need to decide where to offload their tasks. If a user offloads its task to the remote cloud, it will experience transmission delay due to the congestion at the backhaul of the network. On the other hand, if the user offloads its task to edge servers, it will choose the edge server that minimizes its cost of completing

This work was supported by the National Natural Science Foundation of China under Grant No. 62002399 and No. 61972113, the Fundamental Research Funds for the Central Universities, Sun Yat-sen University under Grant No. 2021qntd08, the Basic Research Project of Shenzhen Science and Technology Program under Grant No. JCYJ20180306171800589, and the National Science Foundation of China under Grant No. U20A20159 and No. 61972432. *(Corresponding author: Qian Ma.)*

Z. Chen and Q. Ma are with the School of Intelligent Systems Engineering, Sun Yat-sen University, Shenzhen, Guangdong 518107, China (e-mail: chenzy68@mail2.sysu.edu.cn; maqian25@mail.sysu.edu.cn).

L. Gao is with the School of Electronics and Information Engineering, Harbin Institute of Technology, Shenzhen, China (e-mail: gaol@hit.edu.cn).

X. Chen is with School of Computer Science and Eningeering, Sun Yat-sen University, China (e-mail: chenxu35@mail.sysu.edu.cn).

ISBN 978-3-903176-37-9

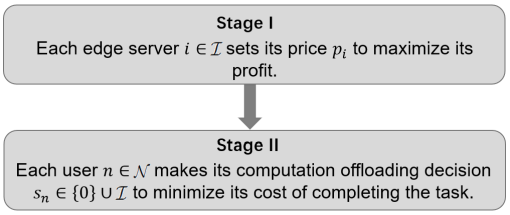

Fig. 2. Two-stage game

the task. Specifically, users need to pay to edge servers for using their computation resources. Besides, users will experience computation delay due to the limited computation resource on each edge server. Furthermore, when a large number of users choose the same edge server simultaneously, users will share the computation resource on the edge server which may cause congestion. In summary, each user selfishly makes its computation offloading decision to minimize its cost of completing the task considering the congestion caused by other users. This motivates us to address the first fundamental question in multi-server edge computing networks:

Key Question 1: *How do mobile users selfishly make their computation offloading decisions?*

Considering users' selfish computation offloading behaviors, edge servers decide the prices charged to users. Since different edge servers usually belong to different business entities, they will compete for market shares to maximize their own profits. We consider the competition among edge servers with heterogeneous marginal service costs due to, for example, different power consumption levels or different operating costs. The heterogeneity of edge servers makes the price competition quite different from the traditional Bertrand competition [7]. Moreover, the congestion effect due to resource sharing among users further complicates the price competition. Specifically, if an edge server sets a low price, a large number of users will choose to offload tasks to the low-priced edge server, which leads to a high congestion level. This will increase users' task completion time, and hence discourage users from choosing this edge server. Therefore, edge servers need to carefully choose their prices to control the congestion levels and users' offloading decisions. Furthermore, the existence of the remote cloud also affects edge servers' price competition. If the prices charged by edge servers are high, users can choose to offload their tasks to the remote cloud. In summary, this motivates us to address the second fundamental question in multi-server edge computing networks:

Key Question 2: *How do heterogeneous edge servers set their prices to maximize their profits?*

As far as we know, we are the first to make the comprehensive economic analysis for multi-server edge computing networks. We model the interactions between edge servers and users as a two-stage game as shown in Fig. 2. In Stage I, each edge server determines its service price to maximize its profit. In Stage II, each user makes its computation offloading decision to minimize its cost of completing the task. We list the main contributions of this paper as follows:

- *Novel Game Analysis in Multi-Server Edge Computing Networks:* To the best of our knowledge, this is the first paper that analyzes the price competition among heterogeneous edge servers considering users' selfish computation offloading behaviors in multi-server edge computing networks through a game-theoretic approach. The heterogeneity of edge servers and the congestion effect due to resource sharing among users complicate the equilibrium analysis.
- *Users' Selfish Computation Offloading:* We model users' behaviors in Stage II as a selfish computation offloading game, which is a population game. By carefully analyzing the structural property of the game, we show that it is a potential game. We discuss its Nash equilibrium (NE) in two cases depending on whether the prices charged by edge servers are high such that some users offload tasks to the remote cloud at equilibrium. We prove the uniqueness of the NE in both cases and derive the explicit expression of the NE.
- *Price Competition among Heterogeneous Edge Servers:* We model the behaviors of heterogeneous edge servers in Stage I as a price competition game. Similarly, we discuss the NE in two cases depending on whether edge servers have high marginal costs. We characterize the conditions for the uniqueness of the NE and derive its explicit expression.
- *Practical Insights:* Our equilibrium analysis helps understand how users make their computation offloading decisions, which facilitates edge servers to better decide their prices. Specifically, users only choose low-priced edge servers at equilibrium, and hence edge servers with low marginal costs can win the price competition. Furthermore, it is surprising that the equilibrium prices do not monotonically increase with the task execution delay.

We organize the rest of the paper as follows. We review the related work in Section II and introduce the system model in Section III. In Section IV, we analyze users' selfish computation offloading game. In Section V, we analyze edge servers' price competition game. We show simulation results in Section VI and conclude in Section VII.

II. Related Work

Existing literature has extensively studied the computation offloading problem in edge computing. In this following, we review literature regarding the optimization-oriented computation offloading and the equilibrium-oriented computation offloading, respectively.

A. Optimization-oriented Computation Offloading

A rich body of literature studies the optimization-oriented computation offloading problem with different optimization goals for different application scenarios [8] [9]. For example, Ma *et al.* in [10] considered the impact of service caching on computation offloading and jointly optimized these two decisions to minimize the overall outsourcing traffic to the cloud. Zhao *et al.* in [11] designed an offloading scheme for dependent tasks that are executed in some order. Xu *et al.*

ISBN 978-3-903176-37-9 © 2021 IFIP

in [12] proposed a distributed algorithm to optimize caching and offloading strategies to minimize the average delay in a long period of time. Li *et al.* in [13] optimized the cost of task offloading while satisfying the service delay constraints for tasks.

The papers for optimization-oriented computation offloading usually aim to optimize the overall system performance from the network operator's point of view, and need a central controller to schedule computation tasks. In our paper, however, we consider the multi-server edge computing networks where each edge server aims to optimize its own benefit.

B. Equilibrium-oriented Computation Offloading

Few works study the equilibrium-oriented task offloading problem where players selfishly optimize their own benefits. Chen *et al.* in [14] studied users' channel selection problem for computation offloading in a multi-channel interference environment and proposed a distributed approach to solve it. Yan *et al.* in [15] proposed a two-stage dynamic game to model and analyze users' offloading decisions as well as the edge server's caching decision and pricing strategy for each service. Since they consider one edge server, the offloading decision is simply a binary variable. References [3] and [16] studied users' computation offloading problem in a multi-server network. Specifically, Apostolopoulos *et al.* in [3] analyzed users' risk-seeking or loss-aversion behaviors and Zhang *et al.* in [16] focused on the load balancing of multiple servers. Furthermore, some papers, as summarized in a recent survey [17], studied the computing resource allocation problem in edge computing through the auction approach. However, no prior work performs a comprehensive economic analysis for multi-server edge computing networks.

In this paper, we focus on the price competition among heterogeneous edge servers in a multi-server edge computing network, considering users' selfish computation offloading behaviors. This problem has not been studied in the existing literature. Our model incorporates the heterogeneity of edge servers, the congestion effect due to resource sharing among users, and the existence of the remote cloud, which makes our model and the derived insights more practically significant.

III. System Model

In this section, we introduce the system model as shown in Fig. 1. We consider a multi-server edge computing network with a remote cloud and a set $\mathcal{I} = \{1, 2, \ldots, I\}$ of edge servers in a densely populated area with a large number of users in set $\mathcal{N} = \{1, 2, \ldots, N\}$. Each user needs to complete a delay-sensitive and computation-intensive application task (e.g., autonomous driving or interactive gaming on VR platforms). Users can offload their tasks to the remote cloud or one of the edge servers, where each edge server sets a price for executing users' tasks.

We model the interactions between edge servers and users as a two-stage game, as shown in Fig. 2. In the following, we first introduce users' selfish computation offloading problem in Stage II, and then present edge servers' price competition problem in Stage I.

A. Users' Selfish Computation Offloading Problem

Each user makes its computation offloading decision to minimize its cost of completing the task. We denote $s_n \in \{0, 1, \ldots, I\}$ as the computation offloading decision of user $n \in \mathcal{N}$. Specifically, $s_n = 0$ indicates that user n offloads its task to the remote cloud, and $s_n = i, \forall i \in \mathcal{I}$ indicates that user n offloads its task to edge server i. We denote user n's task by $\mathcal{T}_n = (b_n, d_n)$. Specifically, b_n represents the size of the input data (in bits) required by the task (e.g., the input figures or the machine learning models for the target detection task), and d_n represents the computation workload (in CPU cycles) of performing the task. For simplicity of analysis, we assume that each user needs to complete the same task (e.g., target detection in autonomous driving) [18], and hence $b_n = b$ and $d_n = d$ for all $n \in \mathcal{N}$.

Next, we first introduce users' costs of completing tasks, and then formulate users' selfish computation offloading game.

1) Users' Cost Model: Users experience different costs when offloading their tasks to edge servers and the remote cloud. In the following, we first model users' costs when offloading tasks to edge servers, and then model users' costs when offloading tasks to the remote cloud.

Offloading Tasks to Edge Servers: When a user offloads its task to edge server $i \in \mathcal{I}$, its cost of completing the task is mainly due to the price p_i charged by the edge server and the delay of executing the task. For the task execution delay, since the edge server is close to the user, the data transmission from the user to the edge server does not suffer from the long transmission delay at the backhaul of the network, and hence the transmission delay is negligible [3]. Therefore, we focus on the computation delay due to the limited computation resource on the edge server.

The computation delay on edge server i depends on the computation capacity f_i (in CPU cycles per second) of edge server i and the amount of tasks offloaded to edge server i. Since we consider the interactions among a large number of users in a densely populated area, the amount of tasks offloaded to each edge server depends on the distribution of users' offloading strategies. Specifically, given the computation offloading strategy profile $\boldsymbol{s} = \{s_n : \forall n \in \mathcal{N}\}$, the proportion of the user population offloading tasks to edge server i is

$$x_i = \frac{\sum_{n \in \mathcal{N}} \mathbb{1}_{\{s_n = i\}}}{N}, \tag{1}$$

where $\mathbb{1}_{\{s_n=i\}} = 1$ if $s_n = i$, and $\mathbb{1}_{\{s_n=i\}} = 0$ if $s_n \neq i$. We denote the population state under the computation offloading strategy profile $\boldsymbol{s}$ by $\boldsymbol{x} = \{x_i : \forall i \in \{0\} \cup \mathcal{I}\}$. Note that when the number of users N is large, the computation offloading strategy of one user does not affect the population state [19].

Under the population state $\boldsymbol{x}$, the computation delay [14] [15] of user n who offloads its task to edge server i, i.e., $s_n = i$, is

$$t^e(s_n = i, \boldsymbol{x}) = \frac{x_i N d}{f_i}. \tag{2}$$

ISBN 978-3-903176-37-9 © 2021 IFIP

Recall that d is the computation workload of a task. Eq. (2) models the fact that users who offload tasks to the same edge server share the computation resource together. Therefore, when a large number of users choose the same edge server, congestion occurs and leads to a long computation delay. For simplicity of analysis, we assume that all edge servers have the same computation capacity, i.e., $f_i = f, \forall i \in \mathcal{I}$ [20]. In this case, we denote $T^E \triangleq \frac{Nd}{f}$, which is the maximum computation delay incurred on an edge server when all users offload tasks to the edge server. We can calculate the computation delay on edge server i as

$$t^e(s_n = i, \boldsymbol{x}) = x_i T^E. \tag{3}$$

We define the total cost [14] of completing the task on edge server $i \in \mathcal{I}$ as

$$F(s_n = i, \boldsymbol{x}) = p_i + \lambda x_i T^E, \tag{4}$$

where λ denotes users' sensitivity to delay. A larger λ indicates that users are more sensitive to the delay of executing the task.

Offloading Tasks to the Remote Cloud: When a user offloads its task to the remote cloud, its cost of completing the task is mainly due to the delay of executing the task. We assume that the remote cloud does not charge users for executing tasks. For example, Tesla provides self-driving services to its car owners and does not charge extra fees for executing the self-driving tasks [21]. For the task execution delay, since the remote cloud usually has ample computation resources, the computation delay is negligible [10]. Therefore, the task execution delay is mainly due to transmitting the input data of the task from the user to the remote cloud through the backbone network. We denote such latency as a constant T^C [22]. In this case, we define the cost of completing the task on the remote cloud as

$$F(s_n = 0, \boldsymbol{x}) = \lambda T^C. \tag{5}$$

2) Selfish Computation Offloading Game: Since each user's cost of completing the task depends on not only its own offloading decision but also other users' decisions, we formulate users' computation offloading problem as a selfish computation offloading game. Since we consider the interactions among a large number of users, the selfish computation offloading game can be modeled as a population game [23] as follows, where each user's total cost depends on its own strategy and the distribution of other users' strategies.

Game 1 (Users' Selfish Computation Offloading Game in Stage II):

- Players: the set $\mathcal{N}$ of users.
- Strategies: Each user $n \in \mathcal{N}$ chooses its computation offloading strategy $s_n \in \{0, 1, \ldots, I\}$.
- Population state: The population state is represented by the vector $\boldsymbol{x} = \{x_0, x_1, \ldots, x_I\}$, where x_i is the proportion of the user population choosing edge server i as calculated in (1).
- Objectives: Each user $n \in \mathcal{N}$ aims to minimize its total cost $F(s_n, \boldsymbol{x})$.

In Game 1, each user $n \in \mathcal{N}$ selfishly makes its computation offloading decision to minimize its own cost, given the population state $\boldsymbol{x}$. Specifically, given the population state $\boldsymbol{x}$, the *best response* of each user $n \in \mathcal{N}$ is defined as

$$BR_n(\boldsymbol{x}) = \arg \min_{s_n \in \{0\} \cup \mathcal{I}} F(s_n, \boldsymbol{x}). \tag{6}$$

We next define the Nash equilibrium (NE) of Game 1.

Definition 1 (Nash Equilibrium of Game 1): The population state $\boldsymbol{x}^*$ is a Nash equilibrium of Game 1 if and only if the strategy of each user belongs to its best response under $\boldsymbol{x}^*$, i,e., $s_n \in BR_n(\boldsymbol{x}^*), \forall n \in \mathcal{N}$.

The Nash equilibrium is a population state under which each user's strategy is the best response to the population state. We will analyze the NE of Game 1 in Section IV.

B. Edge Servers' Price Competition Problem

In this subsection, we first model the profit of each edge server, and then formulate the interactions among edge servers as a price competition game.

1) Edge Servers' Profit Model: Edge servers in set $\mathcal{I}$ belong to different business entities, and they will compete for users through price competition. Specifically, each edge server $i \in \mathcal{I}$ sets a price p_i for executing users' tasks. Furthermore, edge servers are heterogeneous in their marginal service costs (e.g., power consumption levels or operating costs). We denote the marginal cost of edge server $i \in \mathcal{I}$ for serving a user as c_i. In this case, we can calculate the profit of each edge server $i \in \mathcal{I}$ as

$$H_i(p_i, \boldsymbol{p}_{-i}) = (p_i - c_i) x_i(p_i, \boldsymbol{p}_{-i}) N. \tag{7}$$

Here $\boldsymbol{p}_{-i} = \{p_1, \ldots, p_{i-1}, p_{i+1}, \ldots, p_I\}$. Note that the proportion of the user population $x_i(p_i, \boldsymbol{p}_{-i})$ choosing edge server i is the result of the price competition among edge servers, and hence depends on the pricing strategies of all edge servers.

2) Price Competition Game: We model the competition among edge servers as a price competition game as follows.

Game 2 (Price Competition Game in Stage I):

- Players: the set $\mathcal{I}$ of edge servers.
- Strategies: Each edge server $i \in \mathcal{I}$ decides its price p_i charged to users.
- Payoffs: Each edge server $i \in \mathcal{I}$ aims to maximize its profit $H_i(p_i, \boldsymbol{p}_{-i})$ calculated in (7).

Next we define the Nash equilibrium of Game 2.

Definition 2 (Nash Equilibrium of Game 2): A price profile $\boldsymbol{p}^* = \{p_i^* : \forall i \in \mathcal{I}\}$ is a Nash equilibrium of Game 2 if for each edge server $i \in \mathcal{I}$,

$$H_i(p_i^*, \boldsymbol{p}_{-i}^*) \geq H_i(p_i, \boldsymbol{p}_{-i}^*), \text{ for all } p_i \geq c_i. \tag{8}$$

We will solve the two-stage game between edge servers and users through backward induction.

IV. Stage II: Users' Computation Offloading Decisions

In this section, we analyze the Nash equilibrium of users' selfish computation offloading game (i.e., Game 1). Solving the NE of Game 1 by directly analyzing each user's best

ISBN 978-3-903176-37-9 © 2021 IFIP

response is difficult since it is challenging to compute the fixed point of the best response mapping. Furthermore, the congestion effect due to resource sharing among users on the same edge server and the existence of the remote cloud couple users' decisions in a complicated manner.

Instead of directly analyzing users' best response, we will show that users' selfish computation offloading game is a potential game. In this case, all users' costs depend on the potential function, and we can characterize the NE by solving an optimization problem.

Definition 3 (Potential Game): A game is a potential game [24] if there exists a continuously differentiable potential function $G(\boldsymbol{x})$ such that for each feasible population state $\boldsymbol{x}$, $\frac{\partial G}{\partial x_i}(\boldsymbol{x}) = F(i, \boldsymbol{x}), \forall i \in \{0\} \cup \mathcal{I}$.

The key step to identify a potential game is to construct the potential function, which requires careful analysis of the specific structure of the game. Next in Theorem 1, we find a potential function $G(\boldsymbol{x})$ and show that users' selfish computation offloading game is a potential game.

Theorem 1: Users' selfish computation offloading game is a potential game with a potential function

$$G(\boldsymbol{x}) = \sum_{i\in\mathcal{I}} \left(\frac{1}{2}\lambda T^E x_i^2 + p_i x_i\right) + \lambda T^C x_0. \tag{9}$$

Proof See Appendix A in the online technical report [25].

Since users' selfish computation offloading game is a potential game, the Nash equilibrium $\boldsymbol{x}^*$ is the optimal solution to the following optimization problem:

$$\min_{\boldsymbol{x}} \ G(\boldsymbol{x}) \tag{10a}$$

$$\text{subject to } \sum_{i\in\{0\}\cup\mathcal{I}} x_i = 1, \tag{10b}$$

$$x_i \geq 0, \forall i \in \{0\} \cup \mathcal{I}. \tag{10c}$$

Specifically, the Nash equilibrium $\boldsymbol{x}^*$ satisfies the conditions characterized in the following lemma.

Lemma 1: The computation offloading strategy profile $\boldsymbol{x}^*$ is the Nash equilibrium of Game 1 if and only if there exist $\boldsymbol{\alpha} = \{\alpha_i : \forall i \in \{0\} \cup \mathcal{I}\} \in \mathbb{R}^{I+1}$ and $\beta \in \mathbb{R}$ such that:

$$F(i, \boldsymbol{x}^*) = \alpha_i + \beta, \qquad \forall i \in \{0\} \cup \mathcal{I}, \tag{11a}$$

$$\alpha_i x_i^* = 0, \alpha_i \geq 0, \qquad \forall i \in \{0\} \cup \mathcal{I}, \tag{11b}$$

$$\sum_{i\in\{0\}\cup\mathcal{I}} x_i^* = 1, \tag{11c}$$

$$x_i \geq 0, \qquad \forall i \in \{0\} \cup \mathcal{I}. \tag{11d}$$

Proof See Appendix B in the online technical report [25].

By analyzing the conditions in Lemma 1, we next show that the NE of Game 1 is unique. Specifically, we derive the explicit expression for the NE, which falls into two cases depending on whether the prices set by edge servers are high such that some users offload their tasks to the remote cloud at equilibrium. Without loss of generality, we assume that edge servers are ranked in an ascending order of their prices, i.e., $p_1 \leq p_2 \leq \cdots \leq p_I$. Given a price profile $\boldsymbol{p} = \{p_i : \forall i \in \mathcal{I}\}$, we find an edge server i^{th} which satisfies

$$p_{i^{th}} \leq \lambda T^C \leq p_{i^{th}+1}.$$

Theorem 2: Users' selfish computation offloading game admits a unique Nash equilibrium $\boldsymbol{x}^* = \{x_i^* : \forall i \in \{0\} \cup \mathcal{I}\}$, which falls into one of the following two cases.

- Case I: If the price profile $\boldsymbol{p} = \{p_i : \forall i \in \mathcal{I}\}$ set by edge servers satisfies
$$\sum_{i=1}^{i^{th}} p_i + \lambda T^E > i^{th}\lambda T^C, \tag{12}$$
then the NE $\boldsymbol{x}^*$ is
$$x_i^* = \begin{cases} \dfrac{\lambda T^E + \sum_{j=1}^{i^{th}} p_j - i^{th}\lambda T^C}{\lambda T^E}, & \text{if } i = 0, \\ \dfrac{\lambda T^C - p_i}{\lambda T^E}, & \text{if } 1 \leq i \leq i^{th}, \\ 0, & \text{if } i > i^{th}. \end{cases} \tag{13}$$
- Case II: If the price profile $\boldsymbol{p} = \{p_i : \forall i \in \mathcal{I}\}$ set by edge servers satisfies
$$\sum_{i=1}^{i^{th}} p_i + \lambda T^E \leq i^{th}\lambda T^C, \tag{14}$$
we find the edge server $\bar{i}$ which satisfies $\bar{i}p_{\bar{i}} - \sum_{i=1}^{\bar{i}} p_i \leq \lambda T^E \leq \bar{i}p_{\bar{i}+1} - \sum_{i=1}^{\bar{i}} p_i$. In this case, the NE $\boldsymbol{x}^*$ is
$$x_i^* = \begin{cases} \dfrac{\frac{\lambda T^E + \sum_{j=1}^{\bar{i}} p_j}{\bar{i}} - p_i}{\lambda T^E}, & \text{if } 1 \leq i \leq \bar{i}, \\ 0, & \text{if } i = 0 \text{ or } i > \bar{i}. \end{cases} \tag{15}$$

Proof See Appendix C in the online technical report [25].

Theorem 2 shows that when the prices set by edge servers are high (i.e., in Case I where $\sum_{i=1}^{i^{th}} p_i + \lambda T^E > i^{th}\lambda T^C$), there will be a positive proportion x_0^* of the user population offloading tasks to the remote cloud. Intuitively, some users offload tasks to the remote cloud at equilibrium if the average cost of offloading to edge servers (i.e., $(\sum_{i=1}^{i^{th}} p_i + \lambda T^E)/i^{th}$) is higher than the cost of offloading to the remote cloud (i.e., λT^C). We can see that x_0^* increases with the prices $p_i, i \leq i^{th}$ charged by edge servers, and decreases with the task execution delay T^C incurred on the cloud. Furthermore, users who offload tasks to edge servers only choose the edge servers with low prices, i.e., $p_i \leq p_{i^{th}}$. The proportion $x_i^*, i \leq i^{th}$ of the user population offloading tasks to edge server i decreases with the price p_i and the maximum computation delay T^E incurred on edge servers, and increases with the task execution delay T^C incurred on the cloud. No user offloads its task to the edge servers with high prices (i.e., $p_i > p_{i^{th}}$) at equilibrium. The number of edge servers i^{th} to which users offload tasks increases with T^C. That is to say, when the task execution delay incurred on the cloud is long, users choose more edge servers for computation offloading.

When the prices set by edge servers are low (i.e., in Case II where $\sum_{i=1}^{i^{th}} p_i + \lambda T^E \leq i^{th}\lambda T^C$), users only offload tasks to the edge servers with low prices, i.e., $p_i \leq p_{\bar{i}}$. The proportion $x_i^*, i \leq \bar{i}$ of the user population offloading tasks to edge server i decreases with edge server i's price p_i and increases with the prices of other edge servers. No user offloads its task to the remote cloud or the edge servers

ISBN 978-3-903176-37-9 © 2021 IFIP

with high prices (i.e., $p_i > p_{\bar{i}}$) at equilibrium. Besides, the constraint $\bar{i}p_{\bar{i}} - \sum_{i=1}^{\bar{i}} p_i \leq \lambda T^E \leq \bar{i}p_{\bar{i}+1} - \sum_{i=1}^{\bar{i}} p_i$ implies that the number of edge servers $\bar{i}$ to which users offload tasks increases with the maximum computation delay $T^E = Nd/f$. This indicates that when the computation workload d of executing a task is high or the computation capacity f of edge servers is small, users tend to offload tasks to more edge servers to ease the congestion on each edge server.

V. Stage I: Edge Servers' Price Decisions

In this section, we analyze the Nash equilibrium of edge servers' price competition game (i.e., Game 2). The equilibrium analysis of Game 2 is challenging due to the following reasons. First, the heterogeneity of edge servers makes the price competition quite different from the traditional Bertrand competition [7]. Second, the congestion effect due to users' resource sharing and the existence of the remote cloud couple edge servers' price decisions in a complicated way.

Next we will discuss the NE of Game 2 in two cases depending on whether the marginal costs of edge servers are too high such that some users offload their tasks to the remote cloud under the equilibrium prices. Without loss of generality, we assume that edge servers are ranked in an ascending order of their marginal costs, i.e., $c_1 \leq c_2 \leq \cdots \leq c_I$. Later we will show that edge servers' equilibrium prices follow the same order of their marginal costs, i.e., $p_1^* \leq p_2^* \leq \cdots \leq p_I^*$. In the following, we first discuss the case where the marginal costs of edge servers are high. We then discuss the case where the marginal costs of edge servers are low.

A. Case I: Edge Servers Have High Marginal Costs

When edge servers have high marginal costs, they can only charge high prices to users at equilibrium, under which there is a positive proportion of the user population offloading tasks to the remote cloud.

Before characterizing the NE, we first calculate the profit of each edge server in this case. According to Theorem 2, under a price profile $\boldsymbol{p}$, the proportion of the user population offloading tasks to edge server $i \in \mathcal{I}$ at equilibrium is $x_i^* = \max\left\{\frac{\lambda T^C - p_i}{\lambda T^E}, 0\right\}$. Therefore, the profit of edge server i is

$$H_i(p_i, \boldsymbol{p}_{-i}) = N(p_i - c_i)\max\left\{\frac{\lambda T^C - p_i}{\lambda T^E}, 0\right\}. \quad (16)$$

We find an edge server i^c which satisfies

$$c_{i^c} \leq \lambda T^C \leq c_{i^c+1}.$$

The following theorem characterizes the unique NE of Game 2 when edge servers have high marginal costs.

Theorem 3: When the marginal costs of edge servers satisfy

$$\sum_{i=1}^{i^c} c_i > i^c \lambda T^C - 2\lambda T^E, \quad (17)$$

edge servers' price competition game admits a unique Nash equilibrium $\boldsymbol{p}^* = \{p_i^* : \forall i \in \mathcal{I}\}$, which is

$$p_i^* = \begin{cases} \dfrac{c_i + \lambda T^C}{2}, & \text{if } i \leq i^c, \\ c_i, & \text{if } i > i^c. \end{cases} \quad (18)$$

Proof See Appendix D in the online technical report [25].

Theorem 3 shows that when edge servers have high marginal costs that satisfy (17), Game 2 admits a unique NE. Specifically, at equilibrium, the edge server who has a low marginal cost (i.e., $c_i \leq c_{i^c}$) sets its equilibrium price to be higher than its marginal cost (i.e., $p_i^* = \frac{c_i + \lambda T^C}{2} > c_i$) and obtains a positive proportion of the user population (i.e., $x_i^* = \frac{\lambda T^C - c_i}{2\lambda T^E} > 0$). Note that in this case, the price competition is mainly the competition between edge servers and the remote cloud, and hence edge servers' equilibrium prices increase with the task execution delay T^C incurred on the cloud. On the other hand, the edge server who has a high marginal cost (i.e., $c_i > c_{i^c}$) sets its equilibrium price to be equal to its marginal cost (i.e., $p_i^* = c_i$) and obtains no user at equilibrium (i.e., $x_i^* = 0$) due to its high equilibrium price. The number of edge servers i^c to which users offload tasks increases with T^C. That is, when the execution delay incurred on the remote cloud increases, more edge servers have the chance to win the price competition.

B. Case II: Edge Servers Have Low Marginal Costs

When edge servers have low marginal costs, they can charge low prices to users at equilibrium, under which all users offload their tasks to edge servers at equilibrium.

Before characterizing the NE, we first calculate the profit of each edge server in this case. According to Theorem 2, under a price profile $\boldsymbol{p}$, the proportion of the user population offloading tasks to edge server $i \in \mathcal{I}$ at equilibrium is $x_i^* = \max\left\{\frac{\lambda T^E + \sum_{j=1}^{\bar{i}} p_j}{\bar{i}\lambda T^E} - \frac{p_i}{\lambda T^E}, 0\right\}$, where $\bar{i}$ satisfies $\bar{i}p_{\bar{i}} - \sum_{i=1}^{\bar{i}} p_i \leq \lambda T^E \leq \bar{i}p_{\bar{i}+1} - \sum_{i=1}^{i} p_i$. Therefore, the profit of edge server i is

$$H_i(p_i, \boldsymbol{p}_{-i}) = N(p_i - c_i)\max\left\{\frac{\lambda T^E + \sum_{j=1}^{\bar{i}} p_j}{\bar{i}\lambda T^E} - \frac{p_i}{\lambda T^E}, 0\right\}. \quad (19)$$

The following theorem characterizes the NE of Game 2 when edge servers have low marginal costs.

Theorem 4: When the marginal costs of edge servers satisfy

$$\sum_{i=1}^{i^c} c_i \leq i^c \lambda T^C - 2\lambda T^E, \quad (20)$$

the Nash equilibrium $\boldsymbol{p}^* = \{p_i^* : \forall i \in \mathcal{I}\}$ of edge servers' price competition game exists and falls into one of the following two cases.

Case (a): If there exists an edge server $\hat{i}$ such that

$$\frac{\hat{i}-1}{2\hat{i}-1}\left(\hat{i}c_{\hat{i}} - \sum_{j=1}^{\hat{i}} c_j\right) \leq \lambda T^E \leq \frac{\hat{i}-1}{2\hat{i}-1}\left(\hat{i}c_{\hat{i}+1} - \sum_{j=1}^{\hat{i}} c_j\right), \quad (21)$$

the equilibrium price profile $\boldsymbol{p}^*$ is unique and

$$p_i^* = \begin{cases} \dfrac{(\hat{i}-1)c_i + \sum_{j=1}^{\hat{i}} c_j}{2\hat{i}-1} + \dfrac{\lambda T^E}{\hat{i}-1}, & \text{if } i \leq \hat{i}, \\ c_i, & \text{if } i > \hat{i}. \end{cases} \quad (22)$$

ISBN 978-3-903176-37-9 © 2021 IFIP

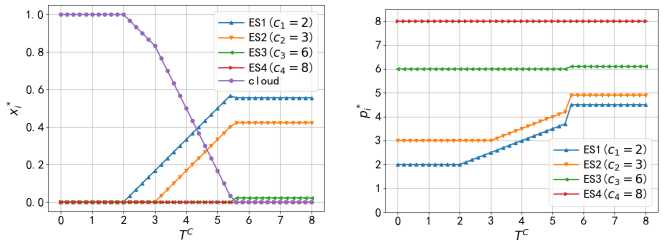

Fig. 3. The equilibrium decisions of users and edge servers under different values of T^C

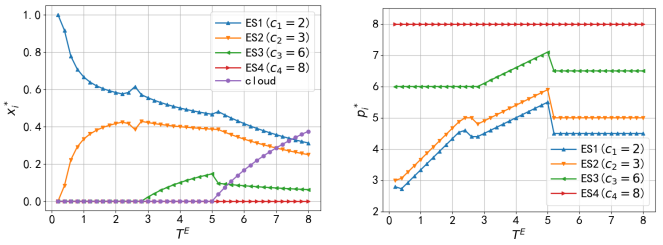

Fig. 4. The equilibrium decisions of users and edge servers under different values of T^E

Case (b): Otherwise, there exists an edge server $\tilde{i}$ such that

$$\frac{\tilde{i}-1}{2\tilde{i}-1}\left(\tilde{i}c_{\tilde{i}+1}-\sum_{j=1}^{\tilde{i}}c_j\right)<\lambda T^E \tag{23}$$
$$<\frac{\tilde{i}}{2\tilde{i}+1}\left((\tilde{i}+1)c_{\tilde{i}+1}-\sum_{j=1}^{\tilde{i}+1}c_j\right).$$

In this case, edge servers' price competition game admits multiple Nash equilibria. Specifically, any price profile $\boldsymbol{p}^*$ that satisfies the following conditions is a NE:

$$\frac{\tilde{i}+1}{2\tilde{i}+1}c_{\tilde{i}+1}+\frac{\tilde{i}}{2\tilde{i}+1}c_i<p_i^* \tag{24}$$
$$<\frac{\tilde{i}}{2\tilde{i}-1}c_{\tilde{i}+1}+\frac{\tilde{i}-1}{2\tilde{i}-1}c_i,\quad 1\le i\le\tilde{i},$$
$$\sum_{i=1}^{\tilde{i}}p_i^*=\tilde{i}c_{\tilde{i}+1}-\lambda T^E, \tag{25}$$
$$p_i^*=c_i,\qquad i>\tilde{i}. \tag{26}$$

Proof See Appendix E in the online technical report [25].

Theorem 4 shows that when edge servers have low marginal costs that satisfy (20), the NE of Game 2 exits but may not be unique. We characterize the condition (21) under which the NE of Game 2 is unique and derive the explicit expression of the NE in (22). Specifically, at equilibrium, the edge servers with low marginal costs (i.e., $c_i \le c_{\tilde{i}}$) win the price competition. Since the price competition is mainly the competition between edge servers with low marginal costs (i.e., $c_i \le c_{\tilde{i}}$), the equilibrium price p_i^* of edge server i where $c_i \le c_{\tilde{i}}$ increases with its own marginal cost c_i and other edge servers' marginal costs $c_j, j \ne i, j \le \hat{i}$. The edge servers who have high marginal costs (i.e., $c_i > c_{\tilde{i}}$) obtain no user (i.e., $x_i^* = 0$) at equilibrium and set the equilibrium prices equal to their marginal costs (i.e., $p_i^* = c_i$).

We then characterize Case (b) where there are multiple Nash equilibria. At equilibrium, edge server i whose marginal cost is lower than $c_{\tilde{i}}$ wins the price competition. Edge servers with marginal costs higher than $c_{\tilde{i}}$ set their prices equal to their marginal costs (i.e., $p_i^* = c_i$) and obtain no user (i.e., $x_i^* = 0$) at equilibrium. Different from the equilibrium in Case (a), the equilibrium price profile in Case (b) can maximize the profits of edge servers $1, 2 \ldots, \tilde{i}$ without introducing edge server $\tilde{i}+1$ to win the market competition. This is guaranteed by (25), under which edge server $\tilde{i}+1$ obtains no user under its equilibrium price, and an arbitrarily small decrease in its price enables it to acquire users. Moreover, the number of edge servers $\tilde{i}$ increases with the maximum computation delay T^E to ease the congestion on each edge server.

VI. SIMULATION RESULTS

In this section, we verify our analysis of the equilibrium behaviors of users and edge servers in the two-stage game through numerical experiments.

We consider a multi-server edge computing network with $I = 4$ edge servers. We assume that $\lambda = 1$, $T^C = 7$, $T^E = 3$, and the marginal costs of 4 edge servers are $\{2, 3, 6, 8\}$ when they are treated as fixed parameters.

We first show the equilibrium decisions of users and edge servers under different values of T^C in Fig. 3. We can see that when $T^C \le 5.4$, at equilibrium of the two-stage game, there is a positive proportion x_0^* of the user population offloading tasks to the remote cloud due to the small task execution delay T^C incurred on the cloud, and x_0^* decreases with T^C, which is consistent with our analysis in Theorem 2. In this case, edge servers with low marginal costs (i.e., $c_i < T^C$) set their equilibrium prices $p_i^* > c_i$ and obtain a positive proportion $x_i^* > 0$ of the user population. Furthermore, p_i^* increases with T^C, which is consistent with our analysis in Theorem 3. When $T^C > 5.4$, at equilibrium of the two-stage game, all users offload their tasks to edge servers due to the large task execution delay T^C incurred on the cloud. In this case, edge servers 1, 2, and 3 who have low marginal costs set their equilibrium prices $p_i^* > c_i$ independent of T^C and obtain a positive proportion $x_i^* > 0$ of the user population, which is consistent with our analysis in Theorem 4.

We then show the equilibrium decisions of users and edge servers under different values of T^E in Fig. 4. We can see that when $T^E \le 5$, at equilibrium of the two-stage game, all users offload their tasks to edge servers due to the small computation delay T^E on edge servers. In this case, edge servers with low marginal costs set their equilibrium prices $p_i^* > c_i$ and obtain a positive proportion $x_i^* > 0$ of the user population. Furthermore, p_i^* generally increases with T^E, which is consistent with our analysis in Case (a) in Theorem 4. Note that when T^E increases from 0.2 to 0.4, edge server 1 decreases its equilibrium price p_1^* due to the competition with edge server 2 which joins the market in this process. And when T^E increases from 2.4 to 2.6, edge servers 1 and 2 decrease their equilibrium prices to prevent edge server 3 from joining the market. When $T^E > 5$, a positive proportion x_0^* of the user population offload tasks to the remote cloud due to the large computation delay T^E on edge servers. Note that x_0^* increases with T^E and x_i^* decreases with T^E, which is

ISBN 978-3-903176-37-9 © 2021 IFIP

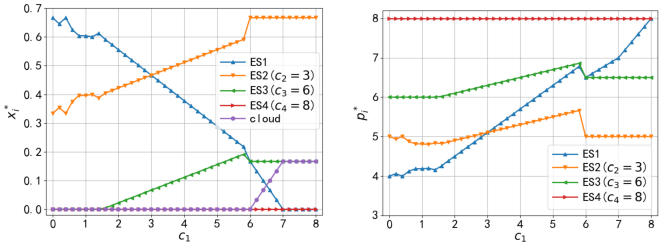

Fig. 5. The equilibrium decisions of users and edge servers under different values of c_1

consistent with our analysis in Theorem 2. In this case, edge servers' equilibrium prices are independent of T^E, which is consistent with our analysis in Theorem 3.

We finally show the equilibrium decisions of users and edge servers under different values of c_1 in Fig. 5. We can see that when $c_1 \leq 1.4$, users only offload their tasks to edge servers 1 and 2 and the sum of their prices keeps unchanged until edge server 3 joins in. It is consistent with the result in Case (b) in Theorem 4. When $1.4 < c_1 \leq 5.8$, all users offload their tasks to edge servers 1, 2 and 3. The equilibrium prices of edge servers 1, 2 and 3 increase with c_1, which is consistent with our analysis in Case (a) in Theorem 4. Furthermore, the proportion x_1^* of users choosing edge server 1 decreases with c_1 due to the increasing equilibrium price p_1^*. However, although p_2^* and p_3^* increase with c_1, the proportions of users x_2^* and x_3^* choosing edge servers 2 and 3 increase with c_1. The reason is that the price increases of edge servers 2 and 3 are lower than the price increase of edge server 1. So even though they increase their prices, they are more competitive than edge server 1. And when $c_1 > 5.8$, some users offload their tasks to the remote cloud. The equilibrium prices sharply decrease due to the remote cloud obtaining some users. Afterwards, the equilibrium price of each edge server only depends on its own cost and the execution delay incurred on the cloud, which is consistent with the analysis in Theorem 3.

VII. Conclusion

In this paper, we study the price competition of heterogeneous edge servers in multi-server edge computing networks, considering users' selfish computation offloading behaviors. We model the interactions between edge servers and users as a two-stage game and analyze its equilibrium. We derive some useful insights that help understand how users make their computation offloading decisions, which facilitates edge servers to better decide their prices. For future work, there are several interesting directions to explore. For example, it would be interesting to study edge servers' caching decisions for heterogeneous tasks and analyze how the caching problem affects edge servers' pricing decisions and users' computation offloading decisions. It is also interesting to analyze the incomplete information scenario where each user does not know the strategies of other users.

References

[1] C.-K. Tham and B. Cao, "Stochastic programming methods for workload assignment in an ad hoc mobile cloud," *IEEE Transactions on Mobile Computing*, vol. 17, no. 7, pp. 1709–1722, 2018.

[2] B. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "Clonecloud: elastic execution between mobile device and cloud," in *EuroSys '11*, 2011.

[3] P. A. Apostolopoulos, E. E. Tsiropoulou, and S. Papavassiliou, "Risk-aware data offloading in multi-server multi-access edge computing environment," *IEEE/ACM Transactions on Networking*, vol. 28, no. 3, pp. 1405–1418, 2020.

[4] Y. Meng, "Idctop ten predictions of china's ict market future companies," 2019. [Online]. Available: http://www.cww.net.cn/article?id=462505

[5] T. Q. Dinh, Q. D. La, T. Q. S. Quek, and H. Shin, "Learning for computation offloading in mobile edge computing," *IEEE Transactions on Communications*, vol. 66, no. 12, pp. 6353–6367, 2018.

[6] Y. Dai, D. Xu, S. Maharjan, and Y. Zhang, "Joint load balancing and offloading in vehicular edge computing and networks," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4377–4387, 2019.

[7] B. Xin and C. Tong, "On a master-slave bertrand game model," *Economic Modelling*, vol. 28, no. 4, pp. 1864–1870, 2011.

[8] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Communications Surveys Tutorials*, vol. 19, no. 3, pp. 1628–1656, 2017.

[9] T. Zheng, J. Wan, J. Zhang, C. Jiang, and G. Jia, "A survey of computation offloading in edge computing," in *2020 International Conference on Computer, Information and Telecommunication Systems (CITS)*, 2020, pp. 1–6.

[10] X. Ma, A. Zhou, S. Zhang, and S. Wang, "Cooperative service caching and workload scheduling in mobile edge computing," in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, 2020, pp. 2076–2085.

[11] G. Zhao, H. Xu, Y. Zhao, C. Qiao, and L. Huang, "Offloading dependent tasks in mobile edge computing with service caching," in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, 2020, pp. 1997–2006.

[12] J. Xu, L. Chen, and P. Zhou, "Joint service caching and task offloading for mobile edge computing in dense networks," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, 2018, pp. 207–215.

[13] R. Li, Z. Zhou, X. Chen, and Q. Ling, "Resource price-aware offloading for edge-cloud collaboration: A two-timescale online control approach," *IEEE Transactions on Cloud Computing*, pp. 1–1, 2019.

[14] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795–2808, 2016.

[15] J. Yan, S. Bi, L. Duan, and Y.-J. A. Zhang, "Pricing-driven service caching and task offloading in mobile edge computing," *IEEE Transactions on Wireless Communications*, pp. 1–1, 2021.

[16] F. Zhang and M. M. Wang, "Stochastic congestion game for load balancing in mobile-edge computing," *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 778–790, 2021.

[17] H. Qiu, K. Zhu, N. C. Luong, C. Yi, D. Niyato, and D. I. Kim, "Applications of auction and mechanism design in edge computing: A survey." *IEEE Communications Surveys Tutorials*, vol. 23, no. 3, 2021.

[18] S. Nath and J. Wu, "Deep reinforcement learning for dynamic computation offloading and resource allocation in cache-assisted mobile edge computing systems," *Intelligent and Converged Networks*, vol. 1, no. 2, pp. 181–198, 2020.

[19] A. P. Kirman, R. J. Aumann, and L. S. Shapley, "Values of non-atomic games," *Economica*, vol. 43, no. 172, p. 445, 2002.

[20] T. Zhao, Z. Sheng, X. Guo, Z. Yun, and Z. Niu, "Pricing policy and computational resource provisioning for delay-aware mobile edge computing," in *IEEE/CIC International Conference on Communications in China*, 2016.

[21] "Tesla owners can try autopilot autonomous driving system for free," 2019. [Online]. Available: http://www.cww.net.cn/article?id=462505

[22] K.-L. Chan, K. Ichikawa, Y. Watashiba, and H. Iida, "Cloud-based vr gaming: Our vision on improving the accessibility of vr gaming," in *2017 International Symposium on Ubiquitous Virtual Reality (ISUVR)*, 2017, pp. 24–25.

[23] J. Barreiro-Gomez, G. Obando, and N. Quijano, "Distributed population dynamics: Optimization and control applications," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 2, pp. 304–314, 2017.

[24] Sandholm, "Potential games with continuous player sets," 1999.

[25] [Online]. Available: https://appendix-1303038105.cos.ap-shenzhen-fsi.myqcloud.com/Appendix.pdf

ISBN 978-3-903176-37-9 © 2021 IFIP