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/TCC.2018.2844379, IEEE Transactions on Cloud Computing

Price-based Resource Allocation for Edge Computing: A Market Equilibrium Approach

Duong Tung Nguyen, *Student Member, IEEE,* Long Bao Le, *Senior Member, IEEE,* and Vijay Bhargava, *Life Fellow, IEEE*

Abstract—The emerging edge computing paradigm promises to deliver superior user experience and enable a wide range of Internet of Things (IoT) applications. In this paper, we propose a new market-based framework for efficiently allocating resources of heterogeneous capacity-limited edge nodes (EN) to multiple competing services at the network edge. By properly pricing the geographically distributed ENs, the proposed framework generates a market equilibrium (ME) solution that not only maximizes the edge computing resource utilization but also allocates optimal resource bundles to the services given their budget constraints. When the utility of a service is defined as the maximum revenue that the service can achieve from its resource allotment, the equilibrium can be computed centrally by solving the Eisenberg-Gale (EG) convex program. We further show that the equilibrium allocation is Pareto-optimal and satisfies desired fairness properties including sharing incentive, proportionality, and envy-freeness. Also, two distributed algorithms, which efficiently converge to an ME, are introduced. When each service aims to maximize its net profit (i.e., revenue minus cost) instead of the revenue, we derive a novel convex optimization problem and rigorously prove that its solution is exactly an ME. Extensive numerical results are presented to validate the effectiveness of the proposed techniques.

Index Terms—Market equilibrium, Fisher market, fairness, algorithmic game theory, edge computing, fog computing.

1 Introduction

The last decade has witnessed an explosion of data traffic over the communication network attributed to the rapidly growing cloud computing and pervasive mobile devices. This trend is expected to continue for the foreseeable future with a whole new generation of applications including 4K/8K UHD video, tactile Internet, virtual/augmented reality (VR/AR), and a variety of IoT applications [1]. As the cloud infrastructure and number of devices continue to expand at an accelerated rate, a tremendous burden will be put on the network. Hence, it is imperative for operators to develop innovative solutions to meet the soaring traffic demand and accommodate diverse requirements of various services and use cases in future networks.

Thanks to the economy of scale and supercomputing capability advantages, cloud computing will likely continue to play a prominent role in the future computing landscape. However, cloud data centers (DC) are often geographically distant from the end-user, which induces enormous network traffic, along with significant communication delay and jitter. Therefore, despite the immense power, cloud computing alone is facing growing limitations in satisfying the stringent requirements in terms of latency, reliability, security, mobility, and localization of new systems and applications (e.g., embedded artificial intelligence, mission-critical communication, 5G wireless systems) [1]. To this end, edge computing (EC) [2], also known as fog computing (FC) [1], has emerged as a novel computing paradigm that complements the cloud and addresses many shortcomings in the traditional cloud model.

In EC, storage, computing, control, and networking resources are placed closer to end-users, things, and sensors. The size of an EN is flexible ranging from smartphones, smart access points (AP), base stations (BS) to edge clouds [3]. For example, a smartphone is the edge between wearable devices and the cloud, a home gateway is the edge between smart appliances and the cloud, a telecom central office is the edge between mobile devices and the core network. By providing elastic resources and intelligence at the edge, EC offers many remarkable capabilities, such as local data processing and analytics, distributed caching, location awareness, resource pooling and scaling, enhanced privacy and security, and reliable connectivity. EC is also a key enabler for ultra-reliable low-latency applications (e.g., AR, autonomous driving). A myriad of benefits and other use cases (e.g., offloading, caching, advertising, healthcare, smart homes/grids/cities) of EC can be found in [1]–[3].

Today, EC is still in the developing stages and presents many new challenges, such as network architecture design, programming models and abstracts, IoT support, service placement, resource provisioning and management, security and privacy, incentive design, and reliability and scalability of edge devices [1]–[3]. In this paper, we focus on the EC resource allocation problem. Unlike cloud computing, where computational capacity of large DCs is virtually unlimited and network delay is high, EC is characterized by relatively low network latency but considerable processing delay due to the limited computing power of ENs. Also, there are a massive number of distributed computing nodes compared to a small number of large DCs. Additionally, ENs may come with different sizes (e.g., number of computing units) and configurations (e.g., computing speed) ranging

- *D.T. Nguyen and V. Bhargava are with the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC, Canada, V6T 1Z4.E-mail: duongnt,vijayb@ece.ubc.ca*
- *L.B. Le is with INRS-EMT, Université du Québec, Montréal, Québec, Canada, H5A 1K6. E-mail: long.le@emt.inrs.ca.*

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/TCC.2018.2844379, IEEE Transactions on Cloud Computing

from a smartphone to an edge cloud with tens/hundreds of servers. These nodes are dispersed in numerous locations with varying network and service delay towards end-users.

On the other hand, different services may have different requirements and properties. Some services can only be handled by ENs satisfying certain criteria. Furthermore, different services may be given different priorities. While every service not only wants to obtain as much resource as possible but also prefers to be served by its closest ENs with low response time, the capacities of ENs are limited. Also, due to the diverse preferences of the services towards the ENs, some nodes can be under-demanded while other are over-demanded. Thus, a fundamental problem is: *given a set of geographically distributed heterogeneous ENs, how can we efficiently allocate their limited computing resources to competing services with different desires and characteristics, considering service priority and fairness?* This work introduces a novel market-based solution framework which aims not only to maximize the resource utilization of the ENs but also to make every service happy with the allocation decision.

The basic idea behind our approach is to assign different prices to resources of different ENs. In particular, highly sought-after resources are priced high while prices of under-demanded resources are low. We assume that each service has a certain budget for resource procurement. The budget can be virtual or real money. Indeed, budget is used to capture service priority/differentiation. It can also be interpreted as the market power of each service. Given the resource prices, each service buys the favorite resource bundle that it can afford. When all the resources are fully allocated, the resulting prices and allocation form a *market equilibrium* (ME). If there is only one EN, an ME can be found easily by adjusting the price gradually until demand equals supply or locating the intersection of the demand and supply curves. However, when there are multiple heterogeneous ENs and multiple services with diverse objectives and different buying power, the problem becomes challenging since the services have more options to buy resources. We consider two distinct market models in this work.

In the first model, the money does not have intrinsic value to the services. Given resource prices, each service aims to maximize its revenue from the allocated resources, without caring about how much it has to pay as long as the total payment does not exceed its budget. This model arises in many real-world scenarios. For example, in 5G networks, the Mobile Edge Computing (MEC) servers of a Telco are shared among different network slices, each of which runs a separate service (e.g., voice, video streaming, AR/VR, connected vehicles, sensing) and serves a group of customers who pay for the service. The Telco can allot different budgets to the slices depending on their importance and/or potential revenue generation (e.g., the total fee paid by the users/subscribers of each slice).

Similarly, an application provider (e.g., Uber, Pokemon Go) or a sensor network may own a number of ENs in a city and need to allocate the edge resources to handle requests of different groups of users/sensors. The budget can be decided based on criteria such as the populations of users/sensors in different areas and/or payment levels (subscription fees) of different groups of users. Another example is that a university (or other organizations) can grant different virtual budgets to different departments or research labs so that they can fairly share the edge servers on the campus. The first model may also emerge in the setting of cloud federation at the edge where several companies (i.e., services) pool their resources together and each of them contributes a fixed portion of resource of every EN. Here, the budgets are proportional to the initial contributions of the companies. Instead of resource pooling, these companies may agree upfront on their individual budgets, and then buy/rent a given set of ENs together.

In these scenarios, it is important to consider both fairness and efficiency. Thus, conventional schemes such as social welfare maximization, maxmin fairness, and auction models may not be suitable. In particular, a welfare maximization allocation often gives most of the resources to users who have high marginal utilities while users with low marginal utilities receive a very small amount of resources, even nothing. Similarly, in auction models, the set of losers are not allocated any resource. Hence, these solutions can be unfair to some users. On the other hands, a maxmin fairness solution often allocates too many resources to users with low marginal utilities, hence, it may not be efficient.

To strive the balance between fairness and efficiency, we advocate the General Equilibrium Theory [4], with a specific focus on the Fisher market model [5], as an effective solution concept for this problem. Specifically, the first model can be cast as a Fisher market in which services act as buyers and ENs act as different goods in the market. For the linear additive utility function as considered in this work, given resource prices, a service may have an infinite set of optimal resource bundles, which renders difficulty in designing distributed algorithms. We suggest several methods to overcome this challenge. Moreover, we show that the obtained allocation is Pareto-optimal, which means there is no other allocation that would make some service better off without making someone else worse off [6]. In other words, there is no strictly "better" allocation. Thus, a Pareto-optimal allocation is efficient.

We furthermore link the ME to the fair division literature [7] and prove that the allocation satisfies remarkable fairness properties including envy-freeness, sharing-incentive, and proportionality, which provides strong incentives for the services to participate in the proposed scheme. Indeed, these properties were rarely investigated explicitly in the ME literature. *Envy-freeness* means that every service prefers its allocation to the allocation of any other service. In an envy-free allocation, every service feels that its share is at least as good as the share of any other service, and thus no service feels envy. *Sharing-incentive* is another well-known fairness concept. It ensures that services get better utilities than what they would get in the *proportional sharing* scheme that gives each service an amount of resource from every EN proportional to its budget. Note that proportional sharing is an intuitive way to share resources fairly in terms of quantity. For the federation setting, sharing-incentive implies that every service gets better off by pooling their resources (or money) together. Finally, it is natural for a service to expect to obtain a utility of at least b/B of the maximum utility that it can achieve by getting all the resources, where b is the payment of the service and B is the total payment of all the services. The *proportionality* property guarantees that the

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/TCC.2018.2844379, IEEE Transactions on Cloud Computing

utility of every service at the ME is at least proportional to its payment/budget. Thus, it makes every service feel fair in terms of the achieved utility.

In the second model, the money does have intrinsic value to the services. The services not only want to maximize their revenues but also want to minimize their payments. In particular, each service aims to maximize the sum of its remaining budget (i.e., surplus) and the revenue from the procured resources, which is equivalent to maximizing the net profit (i.e., revenue minus cost). This model is prevalent in practice. For example, several service providers (SP), each of which has a certain budget, may compete for the available resources of an edge infrastructure provider (e.g., a Telco, a broker). The SPs only pay for their allocated resources and can take back their remaining budgets. Obviously, a SP will only buy a computing unit if the potential gain from that unit outweighs the cost. It is natural for the SPs to maximize their net profits in this case. The classical Fisher market model does not capture this setting since the utility functions of the services depend on the resource prices.

It is worth mentioning that, conventionally, the optimal dual variables associated with the supply demand constraints (i.e., the capacity constraints of the ENs) are often interpreted as the resource prices [32] and common approaches such as network utility maximization (NUM) [33] can be used to compute an ME. However, these approaches do not work for our models that take budget into consideration. Indeed, the main difficulty in computing an ME in both models stems from the budget constraints which contain both the dual variables (i.e., prices) and primal variables (i.e., allocation). In the second model, the prices also appear in the objective functions of the services. Therefore, the ME computation problem becomes challenging. Note that the pair of equilibrium prices and equilibrium allocation has to not only clear the market but also simultaneously maximize the utility of every service (as elaborated in Section 4).

Fortunately, for a wide class of utility functions, the ME in the first model can be found by solving a simple Eisenberg-Gale (EG) convex program [8]–[10]. However, the EG program does not capture the ME in the second model. Interesting, by reverse-engineering the structure of the primal and dual programs in the first model, we can rigorously construct a novel convex optimization problem whose solution is an ME of the second model. This technique can also be used to find the ME that considers other practical constraints (e.g., operation cost of the edge servers). Our main contributions include:

- *Modeling*. We formulate a new market-based EC resource allocation framework and advocate the General Equilibrium theory as an effective solution method for the proposed problem.
- *Centralized solution*. The unique ME in the first model can be determined by the EG program. We also prove some salient fairness features of the ME.
- *Decentralized algorithms*. We introduce several distributed algorithms that efficiently overcome the difficulty raised by the non-unique demand functions of the services and converge to the ME.
- *Extended Fisher market*. We systematically derive a new convex optimization problem whose optimal solution is an exact ME in the extended Fisher market model where buyers value the money.
- *Performance Evaluation*. Simulations are conducted to illustrate the efficacy of the proposed techniques.

The rest of the paper is organized as follows. Section 2 describes related work. The system model and problem formulation are given in Section 3 and Section 4, respectively. The centralized solution using the EG program is analyzed in Section 5. Then, we introduce several distributed algorithms in Section 6. The market model in which buyers aim to maximize their net profits is studied in Section 7. Simulation results are shown in Section 8 followed by conclusions and discussion of future work in Section 9.

2 Related Work

The potential benefits and many technical aspects of EC have been studied extensively in the recent literature. First, the hybrid edge/fog-cloud system can be leveraged to improve the performance of emerging applications such as cloud gaming and healthcare [11], [12]. In [13] , A. Mukherjee *et. al.* present a power and latency aware cloudlet selection strategy for computation offloading in a multi-cloudlet environment. The tradeoff between power consumption and service delay in a fog-cloud system is investigated in [14] where the authors formulate a workload allocation problem to minimize the system energy cost under latency constraints. A latency aware workload offloading scheme in a cloudlet network is formulated in [15] to minimize the average response time for mobile users.

In [16], M. Jia *et. al.* explore the joint optimization of cloudlet placement and user-to-cloudlet assignment to minimize service latency while considering load balancing. A unified service placement and request dispatching framework is presented in [17] to evaluate the tradeoffs between the user access delay and service cost. Reference [18] employs Stackelberg game and matching theory to study the joint optimization among data service operators (DSO), data service subscribers (DSS), and a set of ENs in a three-tier edge network where the DSOs can obtain computing resources from different ENs to serve their DSSs.

Another major line of research has recently focused on the joint allocation of communication and computational resources for task offloading in the MEC environment [19]–[21]. MEC allows mobile devices to offload computational tasks to resource-rich servers located near or at cellular BSs, which could potentially reduce the devices' energy consumption and task execution delay. However, these benefits could be jeopardized if multiple users offload their tasks to MEC servers simultaneously. In this case, a user may not only suffer severe interference but also receive a very small amount of EC resource, which would consequently reduce data rate, increase transmission delay, and cause high task execution time on the servers. Hence, offloading decision, allocation and scheduling of radio resources, and computational resources should be jointly considered in an integrated framework.

Different from the existing literature, which mostly deals with optimizing the overall system performance from a single network operator's point of view, we consider the EC resource allocation problem from the game theory and market design perspectives [8]. In particular, we study how to allocate resources from multiple ENs to multiple services in

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/TCC.2018.2844379, IEEE Transactions on Cloud Computing

a fair and efficient way. We exploit the General Equilibrium [4], a Nobel prize-winning theory, to construct an efficient market-based resource allocation framework. Although this concept was proposed more than 100 years ago [5], only until 1954, the existence of an ME was proved under mild conditions in the seminal work of Arrow and Debreu [4]. However, their proof based on fixed-point theorem is non-constructive and does not give an algorithm to compute an equilibrium [8]. Recently, theoretical computer scientists have expressed great interests in understanding algorithmic aspects of the General Equilibrium concept. Various efficient algorithms and complexity analysis for ME computation have been accomplished over the past decade [8], [22]–[26]. Note that although the existence result has been established, there is no general technique for computing an ME.

Our proposed models are inspired by the Fisher market [5] which is a special case of the exchange market model in the General Equilibrium theory. An *exchange market* model consists of a set of economic agents trading different types of divisible goods. Each agent has an initial endowment of goods and a utility function representing her preferences for the different bundles of goods. Given the goods' prices, every agent sells the initial endowment, and then uses the revenue to buy the best bundle of goods they can afford [4], [8]. The goal of the market is to find the equilibrium prices and allocations that maximize every agent's utility respecting the budget constraint, and the market clears. In the Fisher market model, every agent comes to the market with an initial endowment of money only and wants to buy goods available in the market. We cast the EC resource allocation problem as a Fisher market. We not only show appealing fairness properties of the equilibrium allocation, but also introduce efficient distributed algorithms to find an ME. More importantly, we systematically devise a new and simple convex program to capture the market in which money has intrinsic value to the buyers, which is beyond the scope of the classical Fisher market model.

Indeed, there is a rich literature on cloud resource allocation and pricing [34]. In [35], [36], the authors propose different profit maximization frameworks for cloud providers. References [37]–[39] study how to efficiently share resource and profit among cloud providers in a cloud federation. Several resource procurement mechanisms are introduced in [40] to assist a user to select suitable vendors in a multi-cloud market. In [41], the interaction between a cloud provider and multiple services is modeled as a generalized Nash game. This model is extended to a multi-cloud multi-service environment in [42]. A single-cloud multi-service resource provision and pricing problem with flat, on-demand, and on-spot VM instances is formulated in [43] as a Stackelberg game, which not only maximizes the revenue of the provider but also minimizes costs of the services.

Auction theory has been widely used to study cloud resource allocation [44]–[46]. A typical system consists of one or several clouds and multiple users. First, the users submit bids, which include their desired resource bundles in terms of VM types and quantities as well as the price that they are willing to pay, to an auctioneer. Then, the auctioneer solves a winner determination problem to identify accepted bids. Finally, the auctioneer calculates the payment that each winner needs to pay to ensure truthfulness. In auction, the common objectives are to maximize the social welfare or maximize the profit of the cloud provider. Additionally, only winners receive cloud resources. Furthermore, most of existing auction models do not consider elastic user demands. For example, previous works often assume that cloud users are single-minded, who are interested in a specific bundle only and have zero value for other bundles.

Different from the existing works on cloud economics and resource allocation in general, our design objective is to find a fair and efficient way to allocate resources from multiple nodes (e.g., ENs) to budget-constrained agents (i.e., services), which makes every agent happy with her resource allotment and ensures high edge resource utilization. The proposed model also captures practical aspects, for example, a service request can be served at different ENs and service demands can be defined flexibly rather than fixed bundles as in auction models.

3 System Model

Fig. 1 depicts a generic network architecture that consists of four layers including the traditional cloud layer, the EC platform, the aggregation layer, and the end-device layer. Besides local execution and remote processing at cloud DCs, data and requests from end-devices (e.g., smartphones, set-top-boxes, sensors) can be handled by the EC platform. Note that some data and computing need to be done in the local to keep data privacy. A request typically first goes to a Point of Aggregation (PoA) (e.g., switches/routers, BSs, APs), then it will be routed to an EN for processing. In the EC environment, various sources (e.g., smartphones, PCs, servers in a lab, under-utilized small/medium data centers in schools/hospitals/malls/enterprises, BSs, telecom central offices) can act as ENs. Indeed, service/content/application providers like Google, Netflix, and Facebook can proactively install their content and services onto ENs to serve better their customers. Additionally, enterprises, factories, organizations (e.g., hospitals, universities, museums), commercial buildings (shopping malls, hotels, airports), and other third parties (e.g., sensor networks) can also outsource their services and computation to the intelligent edge network.

We consider a system encompassing various services and a set of geographically distributed ENs with different

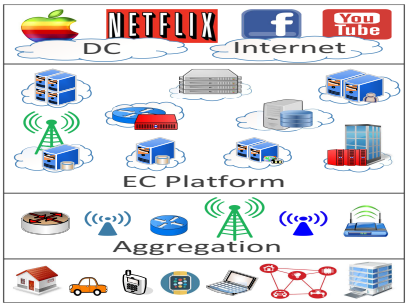


Fig. 1: An EC platform consists of geographically distributed ENs with various configurations. User/service requests are first aggregated at the aggregation layer, then routed to the ENs for processing. Requests that are not handled by the EC platform will be redirected to remote cloud.

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/TCC.2018.2844379, IEEE Transactions on Cloud Computing

configurations and limited computing capacities. Each service has a budget for resource procurement and wants to offload as many requests as possible to the edge network. The value of an EN to a service is measured in terms of the maximum revenue that it can generate by using the EN's resource. An EN may have different values to different services. Since some ENs (e.g., ones with powerful servers) can be over-demanded while some others are under-demanded, it is desirable to harmonize the interests of the services so that each service is happy with its allotment while ensuring high resource utilization. An intuitive solution is to assign prices to ENs and let each service choose its favorite resource bundle. We assume that there is a platform lying between the services and the ENs. Based on the information collected from the ENs (e.g., computing capacity) and the services (e.g., budgets, preferences), the platform computes an ME solution including resource prices and allocation, which not only maximizes the satisfaction of every service but also fully allocates the ENs' resources.

In the first model, each service seeks solely to maximize its revenue under the budget constraint, without concerning about the money surplus after purchasing resources. This can be the case where the services and ENs belong to the same entity, and each service is assigned a virtual budget representing the service's priority. In the second model, the remaining money does have intrinsic value to the services. In this case, each service aims to maximize its net profit. For example, this can be the case where services and ENs are owned by different entities, and each SP (e.g., Google, Facebook, enterprises) has a certain budget for buying resources from an infrastructure provider (e.g., a Telco). For simplicity, we assume that the values of ENs to the services are fixed. Our model can be extended to capture time-varying valuation in a multi-period model by considering each pair of an EN and a time slot as an independent EN.

TABLE 1: NOTATIONS

Notation	Meaning
PoA, SP	Point of Aggregation, service provider
EN, EC	Edge node, edge computing, data center
EG, ME	Eisenberg-Gale, market equilibrium
CES	Constant Elasticity of Substitution
MBB, EF	Maximum bang-per-buck, envy-free index
PropDyn	Proportional Response Dynamics
PropBR	Proportional Sharing Best Response (BR)
i, j	Service index and EN index
M, N	Number of ENs and number of services
$\mathcal{M}$, $\mathcal{N}$	Set of ENs and set of services
$D_i(p)$	Set of ENs giving service i MBB at prices p
B_i	Budget of service i
$a_{i,j}$	Revenue of service i from unit resource of EN j
c_j	Resource capacity of EN j
$d^{\mathrm{n}}_{i,j}$	Network delay between service i and EN j
$d^{\mathrm{p}}_{i,j}$	Processing delay of service i at EN j
$T_i^{\max}$	Maximum delay tolerance of service i
$x_{i,j}$	Resource amount of EN j allocated to service i
x_i	Vector of resources allocated to service i
p_j	Price of one computing unit of EN j
p, $\mathcal{X}$	Resource price vector, resource allocation matrix
$u_i(x_i)$	Revenue function of service i
$U_i(x_i, p)$	Utility function of service i
PR_i	Proportionality ratio of service i

4 Problem Formulation

4.1 EC Resource Allocation Problem

Let $\mathcal{M}$, $\mathcal{N}$, M, and N be the sets of ENs and services, and the numbers of ENs and services, respectively. Denote i as the service index and j as the EN index. We assume that each EN j has c_j homogeneous computing units (e.g., servers) [18]. If an EN has several types of computing units, we can always divide the EN into several clusters, each of which contains only homogeneous units. Then, each cluster can be considered as a separate EN. While the computing units in each EN are homogeneous, different ENs can have different types of computing units. Let $x_{i,j}$ be the number of computing units of EN j allocated to service i. The vector of resources allocated to service i is $x_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,M})$. Finally, define B_i as the budget of service i. Table 1 summarizes important notations used in the paper.

Our goal is to compute an ME including an equilibrium price vector $p = (p_1, p_2, ..., p_M)$, where p_j is price of EN j, and a resource allocation matrix $\mathcal{X}$, in which the element at the ith row and jth column is $x_{i,j}$. The utility $U_i(x_i, p)$ of service i is defined as a function of the amount of resources x_i that it receives and the resource prices p. The capacity constraint of ENs renders: $\sum_{i=1}^{N} x_{i,j} \leq c_j,\ \forall j \in \mathcal{M}$. Without loss of generality, we normalize the capacity of every EN to be 1 (i.e., $c_j = 1,\ \forall j$) and scale related parameters (e.g., price, resource allocation) accordingly. This normalization is just to simplify expressions and equations. Hence, we have: $\sum_{i=1}^{N} x_{i,j} \leq 1,\ \forall j,\ \ x_{i,j} \geq 0,\ \ \forall j$.

Each service is a player in our market game. Given a price vector p, service i aims to maximize its utility $U_i(x_i, p)$ subject to the budget constraint $\sum_j x_{i,j} p_j \leq B_i$.

Definition 4.1. An ME solution (p^*,X^*) needs to satisfy two following conditions:

- *Condition 1*: Given the equilibrium resource price vector $p^* = (p_1^*, p_2^*, ..., p_M^*)$, x_i^* is an optimal resource bundle of service i, for all i, i.e., we have

$$x_i^* = (x_{i,1}^*, \ldots, x_{i,M}^*) \in \underset{x_i \geq 0; \sum_j p_j^* x_{i,j} \leq B_i}{\operatorname{argmax}} U_i(x_i, p^*) \quad (1)$$

- *Condition 2*: All the resources are fully allocated, i.e. , we have: $\sum_i x_{i,j} = 1,\ \ \forall j$.

The first condition can be interpreted as the *user satisfaction condition* while the second condition is often called the *market clearing condition* in Economics [6]. The first condition ensures that the equilibrium allocation x_i^* maximizes the utility of service i at the equilibrium prices p^* considering the user budget constraint. The second condition maximizes the resource utilization of the ENs. It also means the ENs' resources are fully sold in the market, which consequently maximizes the profit of every EN since the equilibrium prices are non-negative. The services are players competing for the limited EC resources, while the platform tries to satisfy the market clearing condition. Prices are used to coordinate the market.

Let $u_i(x_i)$ be the gain/profit/revenue of service i can achieve from the procured resources. We consider two models. In the first model (basic model), every service i wants to maximize $U_i(x_i, p) = u_i(x_i)$ and does not care about how much it has to pay as long as the total payment is under its budget. Here, utility of a service is its revenue. In the second

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/TCC.2018.2844379, IEEE Transactions on Cloud Computing

model, instead of revenue, the services aim to maximize their net profits (i.e., revenue minus cost). The service utility in this model is $U_i(x_i, p) = u_i(x_i) - \sum_j p_j x_{i,j},\ \forall i$. We focus on the first model throughout the paper. The second model is examined in Section 7.

4.2 Service Utility Model

In practice, the services may use different criteria to define $u_i(x_i)$. Our framework takes $u_i(x_i)$ as an input to compute an ME solution. How each service evaluates the ENs is not our focus. While the proposed model is generic, we consider linear functions for the ease of exploring the framework. Let $a_{i,j}$ be the gain of service i from one resource unit of EN j. Then, we have: $u_i(x_i) = \sum_j a_{i,j}x_{i,j},\ \forall i$. Extensions to more general functions will be discussed throughout the paper. In the following, we present an example of how $a_{i,j}$ can be computed. We consider only delay-sensitive services, which are also a main target application of EC. For simplicity, we assume that the transmission bandwidth is sufficiently large and the data size of a request is small (e.g., Apple Siri, Google Voice Search, Google Maps, AR, and Translation). Thus, the data transmission delay (i.e., size/bandwidth) is assumed to be negligible and we consider only propagation delay and processing delay [27], [41].

The total delay of a request of service i from the time a user sends the request to the time she receives a response includes the round-trip delay $d_i^{\mathsf{UE-PoA}}$ between the user and a PoA of the service, the round-trip network delay $d_{i,j}^{\mathsf{n}}$ between the PoA and an EN j hosting the service, and the processing delay at the EN $d_{i,j}^{\mathsf{p}}$. Note that an EN can be located in the same place with a PoA (e.g., a BS). In reality, $d_i^{\mathsf{UE-PoA}}$ is quite small, and we assume it is fixed similar to [15]. In other words, we study the system only *from the aggregation level to the EC platform*. For simplicity, we assume that each service is located at one PoA (e.g., an IoT gateway, a BS, a building). If a service has several PoAs, we need to take sum over all the PoAs to get the total number of requests of the service handled by the EC platform. Denote T_i^{max} as the maximum tolerable delay of service i, we have

$$d_{i,j}^{\mathsf{p}} + d_{i,j}^{\mathsf{n}} \leq T_i^{\mathsf{max}}, \quad \forall i,\ j. \tag{2}$$

Obviously, the maximum number of requests $\lambda_{i,j}^{\mathsf{max}}$ that EN j can process is zero if $d_{i,j}^{\mathsf{n}} \geq T_i^{\mathsf{max}}$.

We model the processing delay at ENs using the widely used M/G/1 queues and assume that the workload is evenly shared among computing units [18], [27], [28], [41]. The average response time $d_{i,j}^{\mathsf{p}}$ of EN j for processing service i can be computed as follows:

$$d_{i,j}^{\mathsf{p}} = \frac{1}{\mu_{i,j} - \frac{\lambda_{i,j}}{x_{i,j}}}, \quad \forall i,\ j, \tag{3}$$

where $\mu_{i,j}$ be the service rate of one computing unit of EN j for handling service i, and $\lambda_{i,j}$ is the request arrival rate (i.e., number of requests per time unit) of service i to EN j. For queue stability, we have $\frac{\lambda_{i,j}}{x_{i,j}} < \mu_{i,j},\ \forall i,\ j$. Otherwise, the queuing delay will be infinite as requests accumulated. From (3), we have

$$\begin{aligned} &\frac{1}{\mu_{i,j} - \frac{\lambda_{i,j}}{x_{i,j}}} \leq T_i^{\mathsf{max}} - d_{i,j}^{\mathsf{n}} \\ \Rightarrow\ &\lambda_{i,j} \leq x_{i,j}\Big(\mu_{i,j} - \frac{1}{T_i^{\mathsf{max}} - d_{i,j}}\Big). \end{aligned} \tag{4}$$

Therefore, if $d_{i,j}^{\mathsf{n}} < T_i^{\mathsf{max}}$, the maximum number of requests that service i can process at EN j is

$$\begin{aligned} \lambda_{i,j}^{\mathsf{max}} &= \max\Big\{x_{i,j}\Big(\mu_{i,j} - \frac{1}{T_i^{\mathsf{max}} - d_{i,j}}\Big),\ 0\Big\} \\ &= x_{i,j}q_{i,j}, \quad \forall i,\ j \end{aligned} \tag{5}$$

where $q_{i,j} = \max\Big\{\Big(\mu_{i,j} - \frac{1}{T_i^{\mathsf{max}} - d_{i,j}}\Big),\ 0\Big\}$. Define a successful request as the request whose total delay is smaller or equal to the maximum delay tolerance. Let r_i be the benefit of successfully serving one request of service i [18]. Then, given $x_{i,j}$ computing units, the revenue of service i is

$$u_{i,j}(x_{i,j}) = r_i q_{i,j} x_{i,j} = a_{i,j}x_{i,j}, \quad \forall i,\ j \tag{6}$$

with $a_{i,j} = r_i q_{i,j}$. Thus, we have

$$u_i(x_i) = \sum_{j=1}^{M} u_{i,j} = \sum_{j=1}^{M} a_{i,j}x_{i,j}, \quad \forall i \tag{7}$$

in which $a_{i,j}$ can be computed beforehand. Note that we implicitly assume the request pool of a service is unlimited. We will discuss later how some assumptions can be relaxed.

Definition 4.2. A function $u(.)$ is *homogeneous of degree* d, where d is a constant, if $u(\alpha x) = \alpha^d u(x),\ \forall\ \alpha > 0$ [8].

From (7), it is easy to verify that $u_i(x_i)$ is a linear function that is homogeneous of degree 1.

Remark: The value of an EN to a service can be defined flexibly. For example, a service may give higher values to ENs in a populated area or ENs with high reliability. A suitable weight can be added to $a_{i,j}$. In the proposed model, each service informs the platform its budget and how much it values different ENs. Based on these information, the platform computes suitable resource allocation satisfying given design objectives. How each service utilizes its allocated resources in the operation stage is not the focus of this work. The key concern of our work is how to harmonize the interests of different services that may have different preferences towards the ENs. Also, we consider only delay-sensitive services to illustrate one way to model the service utility function. It can be justified by the fact that non-delay-sensitive services can be handled effectively by cloud DCs and the precious edge resources can be reserved for important low-latency services. Nevertheless, our model is generic enough to handle other service types as long as we can define the utility of a service as a suitable function of its allocated EC resources. Finally, although we consider computing resources only, the proposed framework can apply to a system in which each service evaluates an EN based on a combination of different resource types of the EN, such as computing, storage, and bandwidth.

5 Centralized Solution

In the first model, each service i aims to maximize $U_i(x_i, p) = u_i(x_i) = \sum_j a_{i,j}x_{i,j}$ subject to the budget constraint $\sum_j p_j x_{i,j} \leq B_i,\ \forall i$. If p is a price vector, the ratio $a_{i,j}/p_j$ is defined as the *bang-per-buck* of EN j to service i, which indicates the utility gained by service i through one unit of money spent on EN j (assuming 0/0 = 0). The *maximum bang-per-buck (MBB)* of service i over the set of ENs is $\alpha_i = \max_j\{a_{i,j}/p_j\}$ [22]. The demand set $D_i(p)$ of service i includes all ENs giving it the MBB value, i.e.,

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/TCC.2018.2844379, IEEE Transactions on Cloud Computing

$D_i(p) = \{j : a_{i,j}/p_j = \alpha_i\},\ \forall i$. Intuitively, to maximize its utility, each service will spend full budget to buy resources from only ENs giving it the MBB. Therefore, a pair (X, p) is an ME if: i) given prices p, service i will exhaust its budget to buy resources only from ENs in $D_i(p)$; and ii) the market clears at prices p. In the following, we will show that the ME in the first model can be inferred from the optimal solution of a convex optimization problem. Also, we will describe some properties of the equilibrium. Specifically, for the case of buyers with linear utilities, the ME can be found by solving the EG convex program given below [8], [9]:

$$\underset{\mathcal{X},u}{\text{maximize}} \sum_{i=1}^{N} B_i \ln u_i \tag{8}$$

subject to

$$u_i = \sum_{j=1}^{M} a_{i,j} x_{i,j}, \quad \forall i \tag{9}$$

$$\sum_{i=1}^{N} x_{i,j} \leq 1, \quad \forall j \tag{10}$$

$$x_{i,j} \geq 0, \quad \forall i,\ j. \tag{11}$$

This problem always has an interior feasible solution by simply setting $x_{i,j} = \epsilon > 0$, for all i and j, where ϵ is sufficiently small such that all constraints (10)-(11) are satisfied with strict inequality. Hence, Slaters condition holds and the the Karush–Kuhn–Tucker (KKT) conditions are necessary and sufficient for optimality [32]. Denote η_i, p_j, and $\nu_{i,j}$ as the dual variables associated with constraints (9), (10), and (11), respectively. We have the Lagrangian

$$L(u, X, \eta, p, \nu) = \sum_i B_i \ln u_i + \sum_j p_j (1 - \sum_i x_{i,j}) \tag{12}$$
$$+ \sum_i \eta_i \Big(\sum_j a_{i,j} x_{i,j} - u_i \Big) + \sum_i \sum_j \nu_{i,j} x_{i,j}.$$

The KKT conditions give

$$\frac{\partial L}{\partial u_i} = \frac{B_i}{u_i} - \eta_i = 0,\ \forall i \tag{13}$$

$$\frac{\partial L}{\partial x_{i,j}} = B_i \frac{a_{i,j}}{u_i} - p_j + \nu_{i,j} = 0, \quad \forall i,\ j \tag{14}$$

$$u_i = \sum_j a_{i,j} x_{i,j},\ \forall i;\ p_j (1 - \sum_i x_{i,j}) = 0, \quad \forall j \tag{15}$$

$$\nu_{i,j} x_{i,j} = 0,\ \forall i,\ j;\ p_j \geq 0,\ \forall j;\ \nu_{i,j} \geq 0,\ \forall i,\ j. \tag{16}$$

We can infer the following

$$\forall i, j : \frac{u_i}{B_i} \leq \frac{a_{i,j}}{p_j} \tag{17}$$

$$\forall i, j : \text{if } x_{i,j} > 0 \Rightarrow \nu_{i,j} = 0 \Rightarrow \frac{u_i}{B_i} = \frac{a_{i,j}}{p_j} \tag{18}$$

$$\forall j : p_j > 0 \Rightarrow \sum_i x_{i,j} = 1;\ \sum_i x_{i,j} < 1 \Rightarrow p_j = 0. \tag{19}$$

The dual variable p_j in the EG program can be interpreted as the price of EN j. Conditions (17) and (18) imply that $x_{i,j} > 0$ if and only if $j \in D_i(p)$, i.e., each service buys resources only from ENs giving it the MBB. This also maximizes $u_i(x_i)$. Note that $u_i/B_i = \alpha_i,\ \forall i$. The following theorem captures some properties of the equilibrium as well as the relationship between the EG program and the ME solution.

Theorem 5.1. The optimal solution to the EG convex program (8)-(11) is an ME. Specifically, the Lagrangian dual variables corresponding to the ENs' capacity constraints (10) are the equilibrium prices. At the equilibrium, the resource allocation not only maximizes the utility but also exhausts the budget of every service. Furthermore, each service purchases resources only from ENs giving its MBB. Additionally, the optimal utilities of the services as well as equilibrium prices are unique.

Proof: Let X^* and u_i^* be the optimal solution to the EG program. Then, X^* and u_i^* need to satisfy the KKT conditions (13)-(19). Denote η^*, p^*, and ν^* as the optimal dual variables. From (14), we have

$$B_i \frac{a_{i,j}}{u_i^*} = p_j^* - \nu_{i,j}^*, \quad \forall i,\ j. \tag{20}$$

Multiplying both sides of (20) by $x_{i,j}^*$ and adding the resulting equalities, we get

$$\frac{B_i}{u_i^*} \sum_j a_{i,j} x_{i,j}^* = \sum_j (p_j^* - \nu_{i,j}^*) x_{i,j}^*, \quad \forall i,\ j. \tag{21}$$

Since $\nu_{i,j}^* x_{i,j}^* = 0, \forall i,\ j$, and $u_i* = \sum_j a_{i,j} x_{i,j}^*,\ \forall i$, equation (21) implies $\sum_j p_j^* x_{i,j}^* = B_i,\ \forall i$. Thus, the optimal solution to the EG program (8)-(11) fully exhausts the budget of every service. Furthermore, as shown above, at the optimality, each service buys resources only from ENs giving its MBB value. In other words, the optimal solution to the EG program maximizes the utility of every service subject to the budget constraint because every service uses all of its money to purchase its MBB resources. This can be inferred from (17) and (18).

We now consider the market clearing condition. From (19), we can observe that resources of ENs with positive price p_j are fully allocated. For ENs with zero prices, their resources can be allocated arbitrarily without affecting the optimal utility of service since the price is zero [8]. Thus, the market clears. Since $(X^*,\ p^*)$ satisfies both conditions of an ME, the optimal solution to the EG program is an ME.

Finally, since the objective function (8) is strictly concave in u_i for all i, the optimal utilities are unique. The uniqueness of equilibrium prices can be inferred from (18). □

From (20), if $p_j^* = 0$, then $\nu_{i,j}^* = 0$ and $a_{i,j} = 0,\ \forall i,\ j$, which means an EN has price of zero only when it is not wanted by all services. We can remove this EN from our system. In the following, we consider only the case where $p_j > 0, \forall j$. Also, it can be shown that Theorem 5.1 is not only applied to linear utilities, but also true for a wider class of homogeneous concave utility functions [10]. Please refer to Appendix D in our technical report [47] for more details.

Next, we study the properties of the equilibrium allocation. First, from (8)-(11), it can be easily verified that the equilibrium allocation is scale-free. It means that it does not matter if service i reports $a_i = (a_{i,1}, \ldots, a_{i,M})$ or $e_i a_i$ for some constant e_i, the allocation that it receives is the same. Also, if a service divides its budget into two parts and acts as two different services with the same original utility function, then the total allocation it obtains from the new ME is equal to the original equilibrium allocation. Furthermore, the equilibrium allocation is not only Pareto-optimal but also possesses many appealing fairness properties such as envy-freeness, sharing incentive, and proportionality.

An allocation X is Pareto-optimal if there does not exist any allocation X' such that $u_i(x_i') \geq u_i(x_i)$ for all i, with

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/TCC.2018.2844379, IEEE Transactions on Cloud Computing

strict inequality holds for at least one i. When budgets are equal, an envy-free allocation $\mathcal{X}$ implies $u_i(x_i) \geq u_i(x_{i'})$ for all i and $i' \in \mathcal{N}$ [7]. Since the budgets can be different, we need to extend this classical definition. An allocation $\mathcal{X}$ is envy-free if $u_i(x_i) \geq u_i(x_{i'}\frac{B_i}{B_{i'}}),\ \forall i,\ i' \in \mathcal{N}$. Let $\hat{x}$ be the allocation where each service receives resource from every EN proportional to its budget, i.e., $\hat{x}_{i,j} = \frac{B_i}{\sum_{i'} B_{i'}}, \forall i,\ j$. The *sharing-incentive* property implies $u_i(x_i) \geq u_i(\hat{x}_i),\ \forall i$. Finally, define the proportionality ratio (PR) of service i as: $PR_i(x_i) = \frac{u_i(x_i)}{u_i(C)}$, in which $u_i(C)$ is the utility of service i when it receives all the resources from the market (i.e., $C = (1, ..., 1), C \in \mathcal{R}^M$). If $PR_i(x_i) \geq \frac{B_i}{\sum_{i'} B_{i'}}$, we say that the allocation $\mathcal{X}$ satisfies the *proportionality* property.

Theorem 5.2. At equilibrium, the allocation is Pareto-optimal and envy-free. It also satisfies the sharing-incentive and proportionality properties.

Proof: Pareto Optimality: We show this by contradiction. Assume allocation X^* is not Pareto-optimal. Then, there exists an allocation X' such that $u_i(x'_i) \geq u_i(x^*_i)$ for all i, and $u_i(x'_i) > u_i(x^*_i)$ for some i. Note that $u_i(x_i) = \sum_j a_{i,j}x_{i,j}$. Consider any feasible allocation X'. Recall the MBB of buyer i is $\alpha_i = \max_j \frac{a_{i,j}}{p_j}$. We have

$$\sum_j x'_{i,j}p_j \geq \sum_j x'_{i,j}\frac{a_{i,j}}{\alpha_i} \geq \sum_j x^*_{i,j}a_{i,j}\frac{1}{\alpha_i} = \sum_j x^*_{i,j}p_j. \quad (22)$$

The second inequality is due to $u_i(x'_i) \geq u_i(x^*_i),\ \forall i$. Thus

$$\sum_j x'_{i,j}p_j \geq B_i, \quad \forall i. \quad (23)$$

Since $u_i(x'_i) > u_i(x^*_i)$ for some i, $\sum_j x'_{i,j}p_j \geq B_i$ for some i. Adding both sides of (23) over all buyers renders

$$\sum_i B_i < \sum_i \sum_j x'_{i,j}p_j = \sum_i x_{i,j} \sum_j p_j \leq \sum_j p_j \quad (24)$$

because $\sum_i x'_{i,j} \leq 1,\ \forall j$ (i.e., the capacity constraints of ENs). However, (24) means the total prices of all the ENs is greater than the total budget of all buyers, which cannot occur. Thus, the equilibrium allocation X^* is Pareto-optimal.

- *Envy-freeness*: To prove that X^* is envy-free, we need to show: $B_{i'}u_i(x^*_i) \geq B_i u_i(x^*_{i'}),\ \forall i,\ i' \in \mathcal{N}$. Let $b_{i,j}$ be the total money that service i spends on EN j. We have

$$\begin{aligned} B_{i'}u_i(x^*_i) &= B_{i'}\sum_j a_{i,j}x^*_{i,j} = B_{i'}\sum_j a_{i,j}\frac{b^*_{i,j}}{p_j} \quad (25)\\ &= B_{i'}\sum_j \frac{a_{i,j}}{p_j}b^*_{i,j} = B'_i\alpha_i\sum_j b^*_{i,j}\\ &= B'_i\alpha_i B_i = B_i\alpha_i\sum_j b^*_{i',j}\\ &\geq B_i\sum_j \frac{a_{i,j}}{p_j}b^*_{i',j} = B_i\sum_j a_{i,j}\frac{b^*_{i',j}}{p_j}\\ &= B_i\sum_j a_{i,j}x^*_{i',j} = B_i u_i(x^*_{i'}),\ \forall i,\ j. \end{aligned}$$

Note that the equalities in the second line of (25) can be inferred from the fact that each buyer only buys resources from ENs in its demand set D_i while the first inequality in the fourth line holds because $\alpha_i \geq \frac{a_{i,j}}{p_j},\ \forall i,\ j$.

- *Proportionality:* From Theorem 5.1, $\sum_i x^*_{i,j} = 1,\ \forall j$. Thus, for linear utilities and the envy-free property, we have

$$\begin{aligned} u_i(C) &= u_i\Big(\sum_i x^*_i\Big) = u_i(x^*_i) + \sum_{i'\neq i} u_i(x^*_{i'}) \quad (26)\\ &\leq u_i(x^*_i) + \sum_{i'\neq i}\frac{B_{i'}}{B_i}u_i(x^*_i) = \frac{\sum_{i'} B_{i'}}{B_i}u_i(x^*_i). \end{aligned}$$

Hence, $u_i(x^*_i) \geq \frac{B_i}{\sum_{i'} B_{i'}}u_i(C), \forall i$.

- *Sharing-incentive:* At the ME (X^*, p^*), no service spends more than its budget. We have

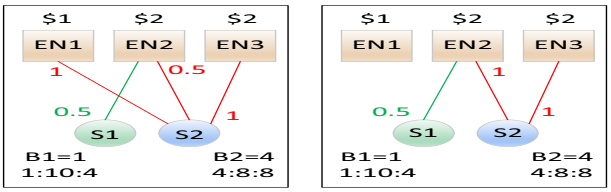
$$\sum_i \sum_j x^*_{i,j}p^*_j \leq \sum_i B_i \Rightarrow \sum_j p^*_j \sum_i x^*_{i,j} \leq \sum_i B_i \quad (27)$$

Thus, $\sum_j p^*_j \leq \sum_i B_i$. Consequently, resource bundle $\hat{x}_i$ costs service i: $\sum_j \hat{x}_{i,j}p^*_j = \sum_j \frac{B_i}{\sum_{i'} B_{i'}}p^*_j \leq B_i, \forall i$. So, service i can afford to buy bundle $\hat{x}_i$ at prices p^*. However, out of all feasible bundles that are affordable to service i, its favorite one is x^*_i. It means $u_i(x^*_i) \geq u_i(\hat{x}_i),\ \forall i$. □

6 Decentralized Solution

A common approach for implementing distributed algorithm is to let the platform iteratively compute prices of the ENs and broadcast the updated prices to the services. Then, each service finds its optimal demand bundle and sends the updated demand to the platform. This price-based strategy can be implemented in a tatonnment style or using the dual decomposition method [33]. Unfortunately, linear utilities may result in non-unique optimal demand bundles because multiple ENs may give the same MBB to a buyer. Hence, the algorithm cannot terminate without aggregated demand coordination from the platform. Consider an example with two services and three ENs. The system parameters are: B_1 = \$1, B_2 = \$4, $a_1 = (1, 10, 4)$, and $a_2 = (4, 8, 8)$.

Fig. 2(a) presents the ME from the centralized EG program. The value associated with each edge between a service and an EN indicates the amount of resource that the service buys from the EN. For example, in Fig. 2(a), we have: $x_{1,1} = 0,\ x_{1,2} = 0.5$, and $x_{1,3} = 0$. The equilibrium price vector is $p = (1, 2, 2)$. The demand sets are: $D_1 = \{2\}$ and $D_2 = \{1, 2, 3\}$. Given the equilibrium prices, the set of optimal (i.e., utility-maximizing) resource bundles of service 2 is infinite. Hence, even if a distributed algorithm reaches the exact equilibrium prices at some iteration, it may not stop since the total demand reported by the buyers may not equal to the total supply. For instance, in Fig. 2(b), although the platform announces the exact equilibrium prices, service 2 may choose to buy all resources from EN2 and EN3. Then, the algorithm may never terminate. In the following, we present two distributed algorithms to find the ME.



(a) With coordination (b) Without coordination

Fig. 2: Market equilibrium with linear utilities

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/TCC.2018.2844379, IEEE Transactions on Cloud Computing

6.1 Dual Decomposition with Function Approximation

Using Lagrangian relaxation [32], [33], we can decompose the EG convex program into sub-problems, each of which can be solved by a service. We observe that the EG program (8)-(11) can be written equivalently as follows.

$$\begin{aligned} \underset{\mathcal{X}}{\text{maximize}} \quad & \sum_{i=1}^{N} B_i \ln u_i(x_i) \\ \text{subject to} \quad & \sum_{i=1}^{N} x_{i,j} \le 1,\ \forall j; \quad x_{i,j} \ge 0,\ \forall i,\ j. \end{aligned} \tag{28}$$

Relaxing the coupling constraints, the partial Lagrangian is

$$\begin{aligned} L(X,p) &= \sum_i B_i \ln u_i(x_i) + \sum_j p_j\big(1 - \sum_i x_{i,j}\big) \\ &= \sum_i \Big(B_i \ln u_i(x_i) - \sum_j p_j x_{i,j}\Big) + \sum_j p_j. \end{aligned} \tag{29}$$

Thus, given a price vector p, each service solves

$$\underset{x_i \ge 0}{\text{maximize}}\ B_i \ln u_i(x_i) - \sum_j p_j x_{i,j}. \tag{30}$$

To overcome the difficulty raised by the non-uniqueness of the optimal demand of the services with linear utilities, we propose to approximate the linear utility function by a Constant Elasticity of Substitution (CES) function, which is widely used in Economics and Computer Science [6], [8]. A CES function has the following form: $u_i^{\mathsf{CES}}(x_i) = \Big(\sum_{j=1}^{M}(a_{i,j}x_{i,j})^{\rho}\Big)^{\frac{1}{\rho}},\ \rho < 1,\ \rho \ne 0$. Indeed, the linear utility function is a special case of the CES function family as $\rho \to 1$. We can approximate the original linear utility function by a CES function where $\rho = 1-\epsilon$ with ϵ is arbitrarily small. As $\epsilon \to 0$, $u_i^{\mathsf{CES}} \to u_i$. Clearly, a CES function is strictly concave and homogeneous [6]. Hence, the EG program and Theorem 5.1 also apply to CES functions [8], [10]. Additionally, we can observe that maximizing a CES function above is equivalent to maximizing $u_i(x_i) = \sum_j (a_{i,j}x_{i,j})^{\rho}$. Since a CES function is strictly concave, the optimal demand bundle of a service is unique. Consider the following optimization problem

$$\underset{x_i \ge 0}{\text{maximize}}\ u_i(x_i) \quad \text{subject to} \ \sum_j p_j x_{i,j} \le B_i. \tag{31}$$

Proposition 6.1. Given a positive price vector p and a CES approximation function, each service i can either solve Problem (30) or Problem (31). Both the problems have the same closed form solution as follows:

$$x_{i,j} = \Big(\frac{a_{i,j}^{\rho}}{p_j}\Big)^{\frac{1}{1-\rho}} \frac{B_i}{\sum_{j=1}^{M}\Big(\frac{a_{i,j}}{p_j}\Big)^{\frac{\rho}{1-\rho}}}. \tag{32}$$

Proof: Refer to our technical report [47]. □

Thus, based on the dual decomposition method where each service solves the sub-problem (30), we have the following distributed algorithm with CES function approximation (**Algorithm 1**). With a sufficiently small step size, it is guaranteed to terminate and converge to an (approximate) global optimal solution [32], [33]. Our simulation results confirm that **Algorithm 1** produces a solution arbitrarily close to the optimal one from the centralized EG program.

Algorithm 1 FUNCTION APPROXIMATION ALGORITHM

1: Initialization: iteration t = 0, set initial prices of ENs $p(0) = p_0$, and set step size $\alpha(0)$ and tolerance γ to be small.
2: **repeat**
3: At iteration t, the platform broadcasts prices p(t) to the buyers.
4: Each buyer computes its optimal demand $x_i(t)$ using (32) and sends it to the platform.
5: The platform updates the prices $p_j(t+1) = \max\Big\{p_j(t) + \alpha(t)\big(1 - \sum_{i=1}^{N} x_{i,j}(t)\big), 0\Big\},\quad \forall j$
6: **until** $\big|p_j(t+1) - p_j(t)\big| < \gamma,\ \forall j$, or the number of iterations t is too large.
7: Output: equilibrium prices p^* and optimal allocation X^*.

6.2 Proportional Response Dynamics Strategy

In this section, we present the Proportional Response Dynamics (**PropDyn**) algorithm proposed by the P2P community. This distributed algorithm is very simple to implement and has been proved to converge to an ME [29]. Basically, in every iteration t, each service updates its bids proportional to the utilities it receives from the previous iteration. Specifically, $b_{i,j}(t) = B_i \frac{u_{i,j}(t-1)}{u_i(t-1)},\ \forall i,\ j,\ t$. Since the ENs' capacities are normalized, the price of an EN equals to the total bids sent to it, i.e., $p_j(t) = \sum_i b_{i,j}(t)$. By bidding $b_{i,j}(t-1)$ to EN j, service i obtains an amount of resource $x_{i,j}(t-1) = b_{i,j}(t-1)/p_j$, and gains a utility $u_{i,j}(t-1) = a_{i,j}x_{i,j}(t-1)$. Finally, $u_i(t-1) = \sum_j u_{i,j}(t-1)$ is the total utility of service i at iteration $t-1$. The salient feature of this algorithm is that it can be implemented efficiently in a distributed manner. In particular, each EN only needs to know the total bid that it receives to compute the price while each buyer only needs to know its own information and learns its utilities achieved in the previous iteration to compute its new bids. The algorithm terminates when the price deviation of every EN is sufficiently small [29]. The major difference between this novel algorithm and traditional distributed algorithms is that in each iteration, every service computes its new bids as mentioned above instead of its optimal demand bundle.

Algorithm 2 BEST RESPONSE DYNAMICS ALGORITHM [30]

1: Sort ENs according to the decreasing order of $\frac{a_{i,j}}{b_{-i,j}}$. Output a sorted list $L_i = \{i_1,\ i_2, \ldots,\ i_M\}$.
2: Find the largest k such that $\frac{\sqrt{a_{i,i_k} b_{-i,i_k}}}{\sum_{j=1}^{k}\sqrt{a_{i,i_j} b_{-i,i_j}}}\Big(B_i + \sum_{j=1}^{k} b_{-i,i_j}\Big) - b_{-i,i_k} \ge 0$
3: Set $b_{i_l} = 0$ for $l > k$, and for $1 \le l \le k$, set $b_{i_l} = \frac{\sqrt{a_{i,i_l} b_{-i,i_l}}}{\sum_{j=1}^{k}\sqrt{a_{i,i_j} b_{-i,i_j}}}\Big(B_i + \sum_{j=1}^{k} b_{-i,i_j}\Big) - b_{-i,i_l}$

To illustrate the effectiveness of the **PropDyn** mechanism as well as the ME concept, we compare it with the *Proportional Sharing Best Response (BR)* mechanism (**PropBR**) proposed in [30], which aims to find a Nash Equilibrium (NE). In a non-cooperative game, a NE is a stable state of a system where no player can gain by a unilateral change of strategy if the strategies of the others are fixed [8]. Both [29] and [30] study a proportional sharing system where the resource of every node is shared proportionally to the services according to their bids. Specifically, we have $x_{i,j} = \frac{b_{i,j}}{b_{i,j}+b_{-i,j}},\ \forall i,\ j$, where $b_{-i,j}$ is the total bid of all the services except i. In both mechanisms, the actions of the services are the bids ($b_{i,j}$) submitted to the ENs. However,

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/TCC.2018.2844379, IEEE Transactions on Cloud Computing

instead of updating its bids following the rule in **PropDyn**, each service in the **PropBR** mechanism selfishly maximizes its utility given strategies taken by other services [30].

Algorithm 2 is the BR algorithm that buyer i will execute given the total bid $b_{-i,j}$ of other buyers. The whole algorithm is implemented in rounds. In each round, each buyer in turns runs **Algorithm 2** and updates its bid vector b_i to the platform. The platform broadcasts new bids to all buyers in the system. A round completes when all buyers have updated their bids. Obviously, whenever this BR dynamics strategy converges, it converges to an NE. As mentioned in [30], the algorithm normally converges after a few rounds.

Interestingly, our simulation shows that buyers do not gain significantly by playing BR. Indeed, most of buyers achieve lower utilities in the **PropBR** scheme compared to the **PropDyn** scheme. Furthermore, to play BR dynamics, each buyer has to know total bids of others and the actual capacity of every EN [30]. In **PropDyn**, buyers only need to know their own information. Therefore, in a proportional sharing system, buyers may not have incentives to play BR.

7 Net Profit Maximization

Different from the **basic model**, in the second model, the services try to optimize their net profits (i.e., revenue minus cost) instead of revenue. Specifically, the net profit of service i is $v_i(x_i) = \sum_j (a_{i,j} - p_j)x_{i,j},\ \forall i$. Given prices p, the objective of service i is to maximize $U_i(x_i, p) = v_i(x_i)$ subject to: $\sum_j x_{i,j}p_j \leq B_i,\ \forall i$ and $x_{i,j} \geq 0,\ \forall i,\ j$. Indeed, maximizing the net profit $v_i(x_i)$ is equivalent to maximizing $\sum_j (a_{i,j} - p_j)x_{i,j} + B_i = \sum_j a_{i,j}x_{i,j} + s_i$, where $s_i = B_i - \sum_j p_j x_{i,j}$ is the surplus money of service i after purchasing x_i. Inspired by the EG program for the **basic model**, we would like to construct a similar convex program to capture the ME in this new model.

Note that without budget consideration, this game-theoretic problem can be solved efficiently by writing down a social welfare maximization problem (i.e., maximizing sum of utilities of all the services), then use the dual decomposition method [33] to decompose it into sub-problems, each of which is solved by one service. Each sub-problem is exactly a net profit maximization problem of a service. Unfortunately, this strategy fails when we consider budget since the social welfare maximization problem cannot be decomposed due to the coupling budget constraints.

Our derivation of the new convex optimization problem is based on reverse-engineering the **basic model**.

Proposition 7.1. The equilibrium prices in the **basic model** can be found by solving the following convex problem.

$$\begin{aligned} \underset{p,\eta}{\text{minimize}} \quad & \sum_{j=1}^{M} p_j - \sum_{i=1}^{N} B_i \ln(\eta_i) \\ \text{subject to} \quad & p_j \geq a_{i,j}\eta_i,\ \forall i,\ j;\ p_j \geq 0,\ \forall j. \end{aligned} \tag{33}$$

Proof: We can obtain this convex problem by using Lagrangian and Fenchel conjugate function [32] to construct the dual problem of the original EG program. Indeed, η_i and p_j are the dual variables associated with (9) and (10). See our technical report for the full proof [47]. □

Clearly, to maximize $v_i(x_i) = \sum_j (a_{i,j} - p_j)x_{i,j}$, service i will never buy resource from EN j if $a_{i,j} < p_j$. In other words, service i would only buy resources from ENs in the set $A_i = \left\{j : \frac{p_j}{a_{i,j}} \leq 1\right\}$. From (33), we have $\eta_i \leq \frac{p_j}{a_{i,j}},\ \forall i$. From these observations, we conjecture that the following prorgram captures the equilibrium prices in our second market model (i.e., net profit maximization).

$$\underset{p,\eta}{\text{minimize}} \quad \sum_{j=1}^{M} p_j - \sum_{i=1}^{N} B_i \ln(\eta_i) \tag{34}$$

subject to

$p_j \geq a_{i,j}\eta_i,\ \forall i,\ j;\ \eta_i \leq 1,\ \forall i;\ \eta_i \geq 0,\ \forall i;\ p_j \geq 0,\ \forall j.$

Theorem 7.2. The solution of the following convex program is exactly an ME of the new market model.

$$\begin{aligned} \underset{\mathcal{X},u,s}{\text{maximize}} \quad & \sum_{i=1}^{N} \left(B_i \ln u_i - s_i\right) \\ \text{subject to} \quad & u_i \leq \sum_{j=1}^{M} a_{i,j}x_{i,j} + s_i, \forall i \\ & \sum_{i=1}^{N} x_{i,j} \leq 1, \forall j; x_{i,j} \geq 0, \forall i, j;\ s_i \geq 0,\ \forall i. \end{aligned} \tag{35}$$

At the equilibrium, the total of money spent and surplus money of every service equals to its budget. Additionally, the optimal utility of every service is unique and greater or equal to its budget. For any buyer who has surplus money, her utility equals her budget.

Proof: See our technical report [47]. □

The convex problem (35) is indeed the dual program of problem (34). We can interpret problem (35) as follows. First, the utility of a service is the sum of its revenue and its surplus money. The first part of the objective function is the weighted sum of logarithmic utilities of the services similar to that of the EG program. However, since the surplus money does not contribute (i.e., not visible) to the market, we should subtract this amount from the aggregated utility function, i.e., the objective function. Finally, similar to the EG program, although budget constraints are not included in (35), the optimal solution satisfies these constraints. It is worth noting that, somewhat surprisingly, although our reverse-engineering approach is specialized for linear revenue functions only, the convex program (35) works also for a wider class of homogeneous concave revenue functions. This proof relies on the fact that if $u_i(x_i)$ is concave and homogeneous of degree one, then $u_i(x_i) + s_i$ is also concave homogeneous of degree one. Please refer to Appendix E in [47] for proof sketch.

8 Numerical Results

8.1 Simulation Settings

We consider a square area with dimensions of 10km x 10km. The locations of ENs and services are generated randomly in the area. We generate a total of 100 ENs and 1000 locations. We assume that each service is located at one location. For the sake of clarity in analysis, in the **base case**, we consider a small system with 8 ENs and 4 services (i.e., M = 8 and N = 4), which are selected randomly in the set of 100 ENs and 1000 services. The network delay between a service and an EN is assumed to be proportional to the distance between

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/TCC.2018.2844379, IEEE Transactions on Cloud Computing

them. The maximum tolerable delay of the services follows a uniform distribution over the interval [15, 25]. The service rate $\mu_{i,j}$ is generated randomly from 80 to 240 requests per time unit. The service price is from 2 to 3 per 100000 requests. The number of computing units in the ENs ranges from 10 to 20. From these parameters, we can compute $a_{i,j}$ of the services as in (6). The net profit maximization model is considered in Section 8.5. In the **base case**, we assume that the services have equal budget. Fig. 3 depicts the valuations of the ENs to the buyers in the **base case**. The **base case** is used in all the simulations unless mentioned otherwise.

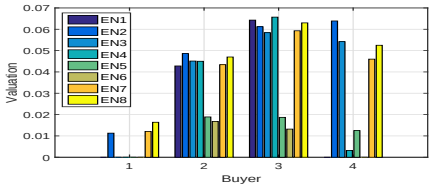
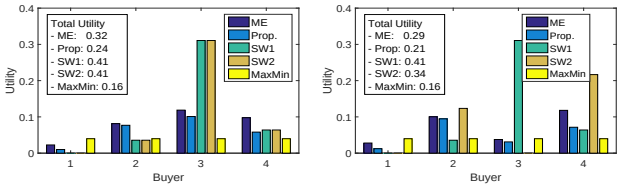


Fig. 3: Valuations of ENs to the buyers

8.2 Performance Comparison

In the first model captured by the EG program, the absolute value of the budget only affects the equilibrium prices by a scaling factor (e.g., all the prices increase twice as the budget of every service is double) and does not affect the allocation and utilities of the services. The budget is normalized such that the total budget of all services is one. The prices act as a means to allocate resources only.

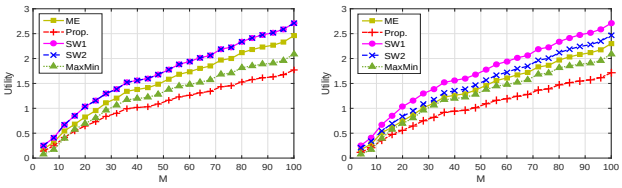


(a) Same budget (b) Budget ratio (4/4/1/4)

Fig. 4: Performance comparison

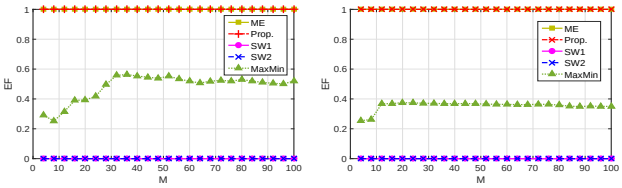
We consider five schemes, including: the proposed *ME*, the proportional sharing (*Prop.*), the social welfare maximization with equal weights (*SW1*), social welfare maximization with different weights (*SW2*), and the maxmin fairness (*MaxMin*) schemes. In the proportional sharing, each buyer i receives $\frac{B_i}{\sum_i B_i}$ portion of resource of every EN. In the social welfare maximization schemes, budget is not considered, and the objective is to maximize $\sum_i w_i u_i(x_i)$ subject to the capacity constraints of the ENs. w_i is the weighting factor of service i. In *SW1*, all weights are equal. In *SW2*, the weight of each service is its budget. Finally, without budget consideration, the *MaxMin* scheme aims to maximize $\min_i u_i(x_i)$ under ENs' capacity constraints.

Figs. 4(a)–6(b) present performance comparison among these schemes under both equal budget and different budget settings. We can observe that the ME scheme balances well the tradeoff between system efficiency and fairness. First, the *ME* scheme considerably outperforms the *Prop.* scheme, which *confirms the sharing-incentive property* of the ME solution. The *MaxMin* scheme produces a fair allocation among the buyers but the total utility of the buyers is much lower compared to other schemes. Noticeably, although the total utility is largest, both schemes *SW1* and *SW2* produce undesirable allocations since some buyers (e.g., buyer 1) are not allocated anything and have zero utility in these schemes. Figs. 6(a), 6(b) compare envy-freeness indices ($EF = \min_{i,j} \frac{u_i(x_i)/B_i}{u_i(x_j)/B_j}$ [30]) of the different schemes. An allocation is envy-free if EF equals to one. The **Prop.** scheme is obviously envy-free by definition. These figures also *confirm that the ME is envy-free*. Furthermore, the proposed ME scheme significantly outperforms the social welfare maximization and *MaxMin* schemes in terms of envy-free fairness. Finally, we can show that the ME satisfies the proportionality fairness property. Due to the space limitation, we leave this result in our technical report [47].



(a) Same budget (b) Budget ratio (4/4/1/4)

Fig. 5: Utility efficiency comparison (N = 4)



(a) Same budget (b) Budget ratio (4/4/1/4)

Fig. 6: Envy-freeness comparison (N = 4)

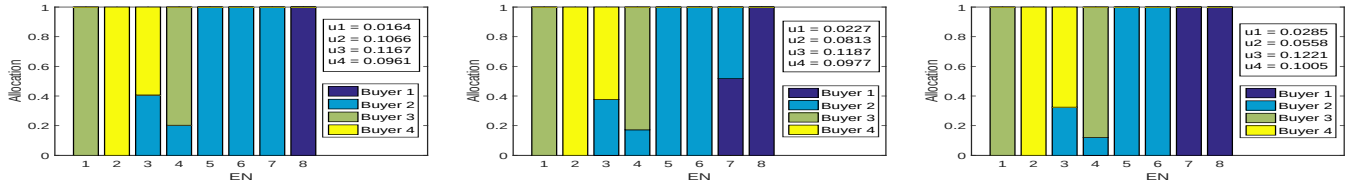
8.3 Sensitivity Analysis

First, we examine the impact of budget on the equilibrium allocation by varying the budget ratio among the buyers. Figs. 7(a)-7(c) show impact of budget on the equilibrium allocation as we vary the budget ratio between services 1 and 2. We observe that buyer 1 is allocated more resources as her budget increases, which also increases her utility. The allocation and utility of buyer 2 decrease as her budget decreases. Fig. 8 further supports this observation where r is the budget ratio between services 1 and 2. Hence, we can conclude that the proposed algorithm is *effective to capture service priority* in terms of budget in the allocation decision.

Fig. 9 shows the dependence of the equilibrium prices on the budget ratio of the buyers. For example, since only EN2, EN7, and EN8 can satisfy the delay requirement of service 1 as seen in Fig. 3, the prices of EN7 and EN8 change considerably as budget of service 1 varies. Also, because EN5 and EN6 are less valuable to the buyers, their equilibrium prices are significantly lower than the prices of other ENs while the prices of EN2 and EN8 are highest because they have high values to all the buyers. These observations imply the proposed method is *effective in pricing*.

The impact of the number of players (i.e., number of ENs and number of services) on the ME is illustrated in Figs. 10(a), 10(b). The buyers have the same budget in this

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/TCC.2018.2844379, IEEE Transactions on Cloud Computing



(a) Budget ratio 0.67/1.33/1/1 (b) Budget ratio 1/1/1/1 (c) Budget ratio 1.33/0.67/1/1

Fig. 7: Impact of budget ratio on the equilibrium allocation

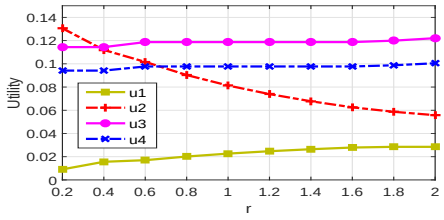


Fig. 8: Impact of budget ratio on the buyers' utilities

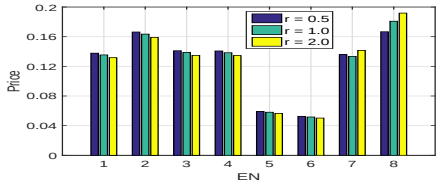
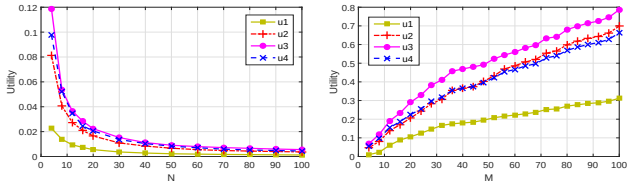


Fig. 9: Impact of budget ratio on the equilibrium prices

case. We show the utilities of buyers 1, 2, 3, and 4 in these figures. As expected, as the number of buyers increases, the utility of individual buyer decreases since the same set of ENs has to be shared among more services. On the other hand, the service utility increases significantly as the number of ENs increases.

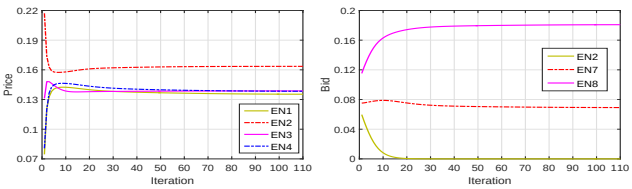


(a) M = 8, varying N (b) N = 4, varying M

Fig. 10: Impact of M and N on the players' utilities

8.4 Analysis of Distributed Algorithms

8.4.1 Proportional Dynamics Allocation



(a) Price trace (b) Buyer 1's bids

Fig. 11: Convergence of EN prices and bids

The proportional dynamics mechanism (**PropDyn**) has low complexity and can be implemented in a distributed manner. The convergence properties of this algorithm in the base case with 8 ENs and 4 services is shown in Figs. 11(a), 11(b). As we can see, the prices and the bids converge after a few tens of iterations. The running time of the algorithm is in order of milliseconds. Figs. 12(a), 12(b), 12(c) compare the buyers' utilities in the **PropDyn** and **PropBR** schemes. We select a particular instance with a set of 10 buyers and 20 ENs from the generated system data. Note that we have run simulation with numerous instances and obtain similar trends. The utility of each buyer in the particular instance is presented in Fig. 12(a). As we can see, the utility values are higher for most of the buyers in the PropDyn scheme compared to the those in the PropBR scheme.

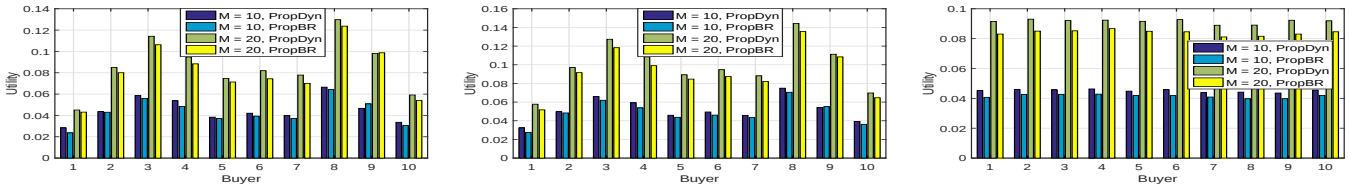
In 12(b), we add a random variable to each $a_{i,j}$ and run the schemes 100 times and take the average results. In 12(c), we generate $a_{i,j}$ randomly in the range between 0.01 and 0.09. As we can observe, the buyers' utilities tend to be higher in the PropDyn scheme in comparison with the PropBR scheme. Furthermore, the PropBR requires buyers to know more system information to play their BR actions in each round. The numerical results show that it brings almost no benefit to the buyers (no utility gain in most cases) to play PropBR scheme. Hence, we can infer that *the buyers should just follow the PropDyn scheme and obtain an ME allocation.*

8.4.2 Function Approximation Algorithm

The convergence properties of the CES approximation scheme as well as its performance are reported. Thanks to the closed form expression of the optimal demand, the algorithm runs very fast even with high number of iterations. As expected, the number of iterations depends strongly on the step size and the initial prices. The convergence of EN6's price (p_6) is shown in Fig. 13(a). The number of iterations decreases as the initial prices are close to the final ME prices, which are unknown. The number of iterations decreases as the step size increases, but we cannot increase the step size γ too much to ensure convergence. Fig. 13(b) presents the price traces of different ENs until convergence with $\alpha = 0.001$ and $p_0 = 0.2$.

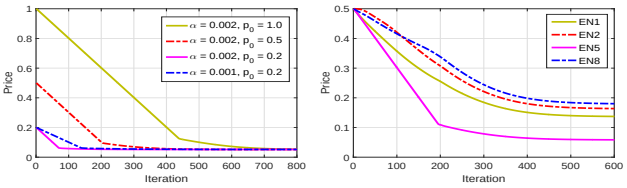
In Fig. 14(a), we study the performance of the approximation scheme by comparing utility of the buyers under the centralized convex program (EG), the approximation CES utility (CES), and the approximation linear utility (**Approx.**). In the *Approx.* scheme, the utility of buyer i is $x_{i,j}a_{i,j}$ where $x_{i,j}$ is the solution of the optimization problem with CES approximation utilities. As we can observe, the values of the utilities are very similar, which confirms that the proposed approximation scheme performs well. In this figure, we set

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/TCC.2018.2844379, IEEE Transactions on Cloud Computing



(a) Utility in one instance (b) Average utility in case 1 (c) Average utility in case 2

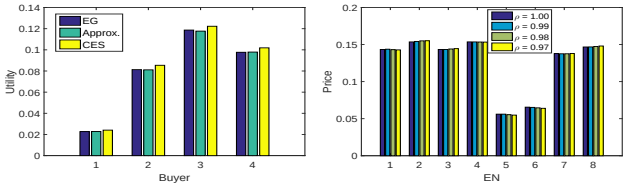
Fig. 12: Utility comparison between PropBR and PropDyn



(a) Number of iterations (p_6) (b) Price convergence

Fig. 13: Convergence of CES approximation

ρ to be 0.99. Finally, the equilibrium prices with different values of ρ is shown in Fig. 14(b). It is easy to see that the prices are almost equal for different values of ρ.



(a) Utility comparison (b) Varying ρ

Fig. 14: CES approximation utility comparison

8.5 Net Profit Maximization Model

We now evaluate the second model where the services aim to maximize their net profits. We use the same system setting with 4 services and 8 ENs in the **base case** as before. From the objective function of the buyer, we know that a buyer will buy resource from an EN only when the price of the EN is less than or equal to its utility gain from the EN. In the revenue maximization case as in the **basic model**, the equilibrium prices increase linearly at the same rate as the budget. However, as can be seen in Fig. 15(a), this property does not hold in the net profit maximization model. Budget scale is the scaling factor by which we multiply the original budget. The figure shows that the equilibrium prices increase then become saturated after certain values of the budgets. At these (saturated) prices, buying resources from the ENs or not does not change the utility for a buyer (i.e., $p_j = a_{i,j}$). When the budget is large enough, the utilities of the buyers become equal to their budgets. It means procuring resources or not does not bring any additional benefit to the buyers. These results are shown in Figs. 15(b), 15(c).

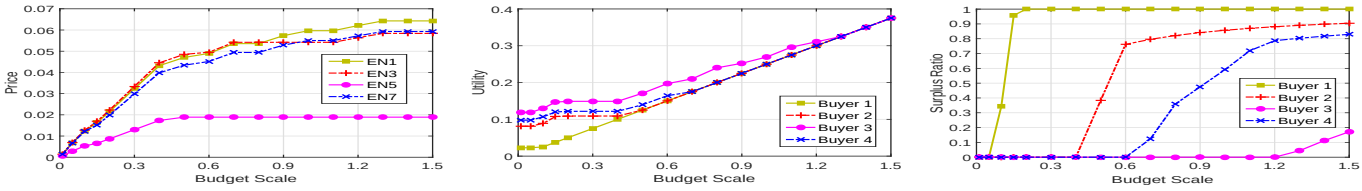
Figs. 16(a), 16(b) present equilibrium prices and optimal utilities, respectively, as the budget varies. **Rev.max** corresponds to the first model (i.e., revenue maximization) with scale equal to 1. As we can observe, for the same budget (i.e., scale = 1), equilibrium prices in the second model are smaller than equilibrium prices in the first model because in the net profit maximization model, a service only buys resource from an EN that gives it positive gain. Also, the service utilities at the equilibrium in the second model is greater than those in the first model due to lower equilibrium prices and budget surplus is considered in the second model. Finally, in the second model, the equilibrium prices and optimal utilities increase as the budget increases. As explained above, equilibrium prices become saturated in the second model at certain points. Hence, the equilibrium prices increase very little as the budget scale increases from 1 to 1.5.

9 Conclusion and Future Works

In this work, we consider the resource allocation for an EC system which consists geographically distributed heterogeneous ENs with different configurations and a collection of services with different desires and buying power. Our main contribution is to suggest the famous concept of General Equilibrium in Economics as an effective solution for the underlying EC resource allocation problem. The proposed solution produces an ME that not only Pareto-efficient but also possesses many attractive fairness properties. The potential of this approach are well beyond EC applications. For example, it can be used to share storage space in edge caches to different service providers. We can also utilize the proposed framework to share resources (e.g., communication, wireless channels) to different users or groups of users (instead of services and service providers). Furthermore, the proposed model can extend to the multi-resource scenario where each buyer needs a combination of different resource types (e.g., storage, bandwidth, and compute) to run its service. We will formally report these cases (e.g., network slicing, NFV chaining applications) in our future work.

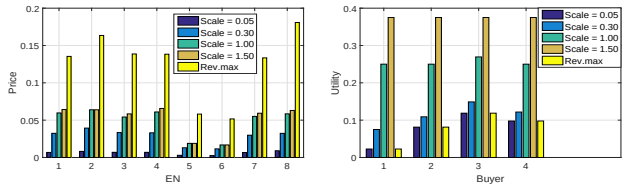
The proposed framework could serve as a first step to understand new business models and unlock the enormous potential of the future EC ecosystem. There are several future research directions. For example, we will investigate the ME concept in the case when several edge networks cooperate with each other to form an edge/fog federation. Investigating the impacts of the strategic behavior on the efficiency of the ME is another interesting topic. Note that N. Chen *et. al.* [24] have shown that the gains of buyers for strategic behavior in Fisher markets are small. Additionally, in this work, we implicitly assume the demand of every service is unlimited. It can be verified that we can add the maximum number of requests constraints to the EG program to capture the limited demand case, and the solution of

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/TCC.2018.2844379, IEEE Transactions on Cloud Computing



(a) Price (b) Utility (c) Surplus Ratio (surplus/budget)

Fig. 15: ME in the net profit maximization model



(a) Prices (b) Utilities

Fig. 16: Impact of budget on equilibrium prices and utilities

this modified problem is indeed an ME. However, although the optimal utilities of the services in this case are unique, there can have infinite number of equilibrium prices. We are investigating this problem in our ongoing work. Also, integrating the operation cost of ENs into the proposed ME framework is a subject of our future work. Finally, how to compute market equilibria with more complex utility functions that capture practical aspects such as task moving expenses among ENs and data privacy is an interesting future research direction. It is also interesting to test the performance of the proposed approach on real datasets of an EC system when EC is widely deployed.

10 ACKNOWLEDGEMENT

This work is supported by the Vanier Canada Graduate Scholarships (Vanier CGS) program.

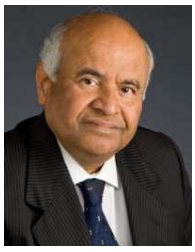
REFERENCES

[1] M. Chiang and T. Zhang, "Fog and IoT: an overview of research opportunities," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 854–864, Dec. 2016.
[2] M. Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, no. 1, pp. 30–39, Jan. 2017.
[3] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.
[4] K.J. Arrow and G. Debreu, "Existence of equilibrium for a competitive economy," *Econometrica*, vol. 22, no. 3, pp. 265–290, 1954.
[5] W.C. Brainard and H.E. Scarf, "How to compute equilibrium prices in 1891," *Cowles Foundation*, Discussion Paper, no. 1272, 2000.
[6] A. Mas-Colell, M. D. Whinston, and J. R. Green, "Microeconomic Theory", 1st ed. New York: Oxford Univ. Press, 1995.
[7] H. Moulin, "Fair division and collective welfare," *MIT Press*, 2004.
[8] N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani, "Algorithmic Game Theory", Cambridge, U.K.: Cambridge Univ. Press, 2007.
[9] E. Eisenberg and D. Gale, "Consensus of subjective probabilities: The pari-mutual method," *Annals of Mathematical Statistics*, vol. 30, pp. 165–168, 1959.
[10] E. Eisenberg, "Aggregation of utility functions," Manage. Sci. 7, PP. 337–350, 1961.
[11] Y. Lin and H. Shen, "CloudFog: leveraging fog to extend cloud gaming for thin-client MMOG with high quality of service," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 2, pp. 431–445, Feb. 2017.
[12] L. Gu, D. Zeng, S. Guo, A. Barnawi, and Y. Xiang, "Cost efficient resource management in fog computing supported medical cyber-physical system," *IEEE Trans. Emerg. Topics Comput.*, vol. 5, no. 1, pp. 108-119, Jan.-Mar. 2017.
[13] A. Mukherjee, D. De, and D.G. Roy, "A power and latency aware cloudlet selection strategy for multi-cloudlet environment," *IEEE Trans. Cloud Comput.* , to appear.
[14] R. Deng, R. Lu, C. Lai, T. H. Luan, and H. Liang, "Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 1171–1181, Dec. 2016.
[15] X. Sun and N. Ansari, "Latency aware workload offloading in the cloudlet network," *IEEE Commun. Lett.*, vol. 21, no. 7, pp. 1481–1484, Jul. 2017.
[16] M. Jia, J. Cao, and W. Liang, "Optimal cloudlet placement and user to cloudlet allocation in wireless metropolitan area networks," *IEEE Trans. Cloud Comput.*, to appear.
[17] L. Yang, J. Cao, G. Liang, and X. Han, "Cost aware service placement and load dispatching in mobile cloud systems," *IEEE Trans. Comput.*, vol. 65, no. 5, pp. 1440–1452, May 2016.
[18] H. Zhang, Y. Xiao, S. Bu, D. Niyato, F.R. Yu, and Z. Han, "Computing resource allocation in three-Tier IoT fog networks: a joint optimization approach combining stackelberg game and matching," *IEEE Internet Things J.*, to appear.
[19] S. Sardellitti, G. Scutari, and S. Barbarossa, "Joint optimization of radio and computational resources for multicell mobile-edge computing," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 1, no. 2, pp. 89–103, Jun. 2015.
[20] X. Lyu, H. Tian, C. Sengul, and P. Zhang, "Multiuser joint task offloading and resource optimization in proximate clouds," *IEEE Trans. Veh. Technol.*, vol. 66, no. 4, pp. 3435–3447, Apr. 2017.
[21] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 4, pp. 974–983, Apr. 2015.
[22] N.R. Devanur, C.H. Papadimitriou, A. Saberi, and V.V. Vazirani, "Market equilibrium via a primal–dual algorithm for a convex program," J. ACM vol. 55, no. 5, article 22, Nov. 2008.
[23] V.V. Vazirani and M. Yannakakis, "Market equilibrium under separable, piecewise-linear, concave utilities," J. ACM, vol. 58, no. 3, article. 10, May 2011.
[24] N. Chen, X. Deng, B. Tang, and H. Zhang, "Incentives for strategic behavior in Fisher market games", in *Proc. Conf. Artificial Intelligence (AAAI)*, pp. 453–459, Phoenix, Arizona, USA, Feb. 2016.
[25] X. Chen, D. Paparas, and M. Yannakakis, "The complexity of non-monotone markets" *J. ACM*, vol. 64, no. 3, Article 20, Jun. 2017.
[26] J. Garg, R. Mehta, M. Sohoni, and V.V. Vazirani, "A complementary pivot algorithm for market equilibrium under separable, piecewise-linear concave utilities", *SIAM J. Comput.*, vol. 44, no. 6, pp. 1820–1847, 2015.
[27] Z. Liu, M. Lin, A. Wierman, S. Low, and L. L. H. Andrew, "Greening geographical load balancing,", *IEEE/ACM Trans. Netw.*, vol. 23, no. 2, pp. 657–671, Apr. 2015.
[28] L. Tang and H. Chen, "Joint pricing and capacity planning in the IaaS cloud market," *IEEE Trans. Cloud Comput.*, vol. 5, no. 1, pp. 57–70, Jan.-Mar. 2017.
[29] F. Wu and L. Zhang, "Proportional response dynamics leads to market equilibrium," in *Proc. the thirty-ninth annual ACM symposium on Theory of Computing (STOC)*, pp. 354–363, New York, NY, USA, 2007.

[30] M. Feldman, K. Lai, and L. Zhang, "The proportional-share allocation market for computational resources," *IEEE Trans. Parallel Distrib. Syst.*, vol. 20, no. 8, pp. 1075–1088, Aug. 2009.
[31] J. F. Nash, "The bargaining problem", *Econometrica 28*, pp. 155–162, 1950.
[32] S. Boyd and L. Vandenberghe, "Convex Optimization", Cambridge, U.K.: Cambridge Univ. Press, 2004.
[33] D.P. Palomar and M. Chiang, "A tutorial on decomposition methods for network utility maximization," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1439–1451, Aug. 2006.
[34] N.C. Luong, P. Wang, D. Niyato, Y. Wen, and Z. Han, "Resource management in cloud networking using economic analysis and pricing models: a survey," *IEEE Commun. Surv. Tut.*, vol. 19, no. 2, pp. 954–1001, Secondquarter 2017.
[35] H. Xu and B. Li, "Dynamic cloud pricing for revenue maximization," *IEEE Trans. Cloud Comput.*, vol. 1, no. 2, pp. 158–171, Jul.–Dec. 2013.
[36] A.N. Toosi, K. Vanmechelen, K. Ramamohanarao, and R. Buyya, "Revenue maximization with optimal capacity control in infrastructure as a service cloud markets," *IEEE Trans. Cloud Comput.*, vol. 3, no. 3, pp. 261–274, Jul.–Sept. 2015.
[37] L. Mashayekhy, M. M. Nejad, and D. Grosu, "Cloud federations in the sky: formation game and mechanism," *IEEE Trans. Cloud Comput.*, vol. 3, no. 1, pp. 14–27, Jan.-Mar. 2015.
[38] M. Hadji and D. Zeghlache, "Mathematical programming approach for revenue maximization in cloud federations," *IEEE Trans. Cloud Comput.*, vol. 5, no. 1, pp. 99–111, Jan.-Mar. 2017.
[39] I. Petri, J. Diaz-Montes, M. Zou, T. Beach, O. Rana, and M. Parashar, "Market models for federated clouds," *IEEE Trans. Cloud Comput.*, vol. 3, no. 3, pp. 398–410, Jul.—Sept. 2015.
[40] A. Prasad and S. Rao, "A mechanism design approach to resource procurement in cloud computing," *IEEE Trans. Comput.*, vol. 63, no. 1, pp. 17–30, Jan. 2014.
[41] D. Ardagna, B. Panicucci, and M. Passacantando, "Generalized Nash equilibria for the service provisioning problem in cloud systems," *IEEE Trans. Serv. Comput.*, vol. 6, no. 4, pp. 429–442, Oct.-Dec. 2013.
[42] D. Ardagna, M. Ciavotta, and M. Passacantando, "Generalized Nash equilibria for the service provisioning problem in multi-cloud systems," *IEEE Trans. Serv. Comput.*, vol. 10, no. 3, pp. 381–395, May-Jun. 2017.
[43] V. Cardellini, V. Di Valerio, and F. Lo Presti, "Game-theoretic resource pricing and provisioning strategies in cloud systems," *IEEE Trans. Serv. Comput.*, to appear.
[44] L. Mashayekhy, M.M. Nejad, and D. Grosu, "Physical machine resource management in clouds: a mechanism design approach," *IEEE Trans. Cloud Comput.*, vol. 3, no. 3, pp. 247–260, Jul.–Sep. 2015.
[45] X. Wang, X. Wang, H. Che, K. Li, M. Huang, and C. Gao, "An intelligent economic approach for dynamic resource allocation in cloud services," *IEEE Trans. Cloud Comput.*, vol. 3, no. 3, pp. 275–289, Jul.–Sept. 2015.
[46] S. Chichin, Q.B. Vo, and R. Kowalczyk, "Towards efficient and truthful market mechanisms for double-sided cloud markets," *IEEE Trans. Serv. Comput.*, vol. 10, no. 1, pp. 37–51, Jan.-Feb. 2017.
[47] Technical report. Available [Online]: *https://arxiv.org/abs/1805.02982*.

Duong Tung Nguyen received the B.Sc. degree in electronics and telecommunications from the Hanoi University of Science and Technology, Hanoi, Vietnam, in 2011, and the M.Sc. degree in telecommunications from the Institut National de la Recherche Scientifique, Université du Québec, Montréal, QC, Canada, in 2014. Currently, he is a PhD student at the Department of Electrical and Computer Engineering, University of British Columbia, Canada.

Long Bao Le (S'04-M'07-SM'12) received the B.Eng. degree from Ho Chi Minh City University of Technology, Vietnam, in 1999, the M.Eng. degree from Asian Institute of Technology, Pathumthani, Thailand, in 2002, and the Ph.D. degree from the University of Manitoba, Winnipeg, MB, Canada, in 2007. Since 2010, he has been with the Institut National de la Recherche Scientifique (INRS), Université du Québec, Montréal, QC, Canada, where he is currently an Associate Professor.



Vijay K. Bhargava (S'70-M'74-SM'82-F'92-LF'13) received the B.A.Sc., M.A.Sc., and Ph.D. degrees from Queens University, Kingston, ON, Canada, in 1970, 1972, and 1974, respectively. Currently, he is a Professor with the Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, BC, Canada. He has served as an Editor-in-Chief of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS (2007–2009). He is a past President of the IEEE Information Theory Society and the IEEE Communications Society.