On Resource Management for Cloud Users: A Generalized Kelly Mechanism Approach

Richard T.B. Ma
Advanced Digital Science Center, Illinois at Singapore Pte. Ltd.
Dah Ming Chiu and John C.S. Lui
Department of IE and CS&E, The Chinese University of Hong Kong
Vishal Misra and Dan Rubenstein
Department of Computer Science, Columbia University

Abstract

Cloud computing provides network software companies with a platform to develop Software as a Service (SaaS) applications. The *pay-as-you-go* pricing model unties these SaaS providers from large capital outlays in hardware deployment and maintenance. Depending on the level of virtualization and service abstraction, cloud vendors provide different functionalities to the SaaS providers. For example, with a high-level service abstraction, Google AppEngine can provide automatic scalability for web applications; however, it cannot support general-purpose applications and does not give service controls for the SaaS providers. On the other hand, Amazon EC2 provides a low-level virtualization that allows for general-purpose applications. Nevertheless, it cannot scale the applications automatically for SaaS providers. Moreover, managing the computing resources for SaaS processes can also be challenging for a SaaS provider. The aim of this paper is to provide a theoretical framework for resource management for Saas providers so they can efficiently control the service levels of their users, and as to easily scale their applications under dynamic user arrivals/departures. Our resource bidding and allocation framework can be viewed as a generalization of the Kelly mechanism. Previous work showed that the efficiency loss of the Kelly mechanism is bounded by 25% of the optimal social welfare. By using a built-in penalty differentiation, our mechanism is able to close the efficiency gap. To achieve this efficiency, a feedback control mechanism is proposed to maximize the aggregate valuation of users.

I. Introduction

Cloud computing vendors, e.g. Amazon and Google, provide computing services as a utility and charge cloud users in a pay-as-you-go manner. By adopting this cloud computing paradigm, network software companies can avoid large capital outlays in hardware deployment and maintenance and provide *Software-as-a-Service* (SaaS) to their users. Through virtualization, cloud providers isolate low-level functionalities and the underlying hardware from cloud users. Based on the level of abstraction at which cloud users can manage resources, we can identify different classes [4] of cloud computing services.

With a high-level service abstraction, Google AppEngine [2] works exclusively for web applications. Although application providers cannot control the resources and service level for their users, AppEngine can automatically scale and provide high-availability for the applications. However, AppEngine's scaling ability relies on the request-reply nature of web applications, which makes resource provisioning easy, it cannot serve general-purpose computing services. Amazon Elastic Compute Cloud (EC2) [1], on the other hand, allows low-level controls over virtualized hardware, e.g. raw CPU cycles and IP-level connectivity. Each EC2 Virtual Machine (VM) instance is specified by number of virtual CPU cores, capacity of RAM and storage. However, Amazon cannot achieve automatic scalability for the applications. Thus, it becomes the cloud users' responsibility and concern to provide resource allocation and service control for their SaaS processes and the scalability for their SaaS applications.

Without a clear resource management model provided by cloud providers, it can be quite difficult for cloud users to manage the virtualized resources so as to achieve service levels and scalability. If resources are over-provisioned, cloud users might still need to pay for the idle instances of virtual machines, e.g. Amazon EC2 instances; if resources are under-provisioned, then too many SaaS processes will be assigned to a single instance of virtual machine and the service quality will deteriorate. With the unpredictable dynamic user demand, the mismatch between resource and service demand will become more severe if the cloud users cannot scale their application accordingly. *To address this problem, we propose a resource management framework under which the cloud users can easily manage the virtualized resources and provide service controls and scalability to their SaaS applications.*

In particular, we design a resource allocation mechanism that generalizes the Kelly mechanism [14]. By carefully choosing a penalty differentiation, it can also close up the efficiency gap of the Kelly mechanism. Our contribution includes:

- We design (Section II-B) a novel build-in penalty differentiation to generalize the Kelly mechanism.
- We analyze (Section III) the resource competition game induced by the generalized mechanism and show:
  - Under any penalty differentiation, the competition game has a unique Nash equilibrium (Thm. 1).
  - Any non-dictatorial resource allocation can be realized as a unique Nash equilibrium (Theorem 2).
  - There exists a bijective mapping between the domain of penalty and resource allocation (Theorem 5).

- We characterize the optimality condition for the social welfare and design a feed-back control mechanism to maximize it (Section IV).
- We further generalize the mechanism for heterogeneous resources for cloud computing services (Section V).

We believe that our new framework provides cloud users with better resource management for their SaaS processes as well as better service and quality controls for their applications.

II. Resource Management Model and Assumptions

A. A Three-party Overview

We use a similar three-party view of Armnrust el. al. [4] to illustrate a common cloud computing architecture. Figure 1 depicts these three parties: *the cloud providers*, *the cloud users* (or the SaaS providers) and *the SaaS users* (or processes).

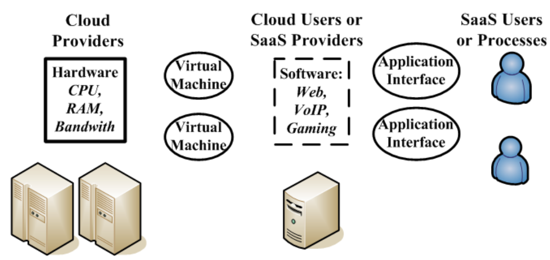


Fig. 1. A three-party view of cloud computing.

Unlike the Google AppEngine that only supports web applications, we assume that the cloud providers support general-purpose cloud computing services, and that the SaaS applications and users are invisible to the cloud providers. Cloud providers allocate VM instances to cloud users on demand, but they do not provide automatic scaling feature for the SaaS applications running on the VMs. We do assume that, for each VM instance, the cloud providers will give an interface for its users to manage various computing resources, e.g. CPU, RAM and bandwidth, for different SaaS processes on the VM instance. Figure 2 illustrates the interaction between a typical cloud user and its cloud provider. Each cloud user decides when and how many VMs to allocate/de-allocate, e.g., the cloud user is allocating two VMs to host four of its SaaS processes. For each VM instance, the cloud user manages its SaaS processes in terms of how much resource each process consumes, e.g. on VM 1, the cloud user decides to give a higher priority to one of the SaaS processes and delivers the resource allocation command to the cloud provider. The cloud provider then executes the resource allocation command without knowing the identities of the SaaS processes.

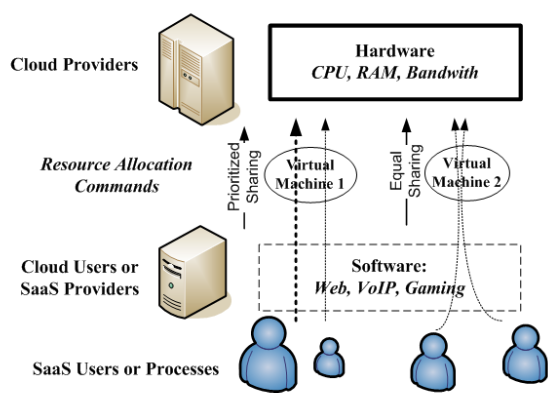


Fig. 2. Interaction between a cloud provider and a cloud user.

The objective of a cloud user is to provide service level controls on its SaaS processes as well as to scale the application based on the demands. Figure 3 illustrates the interaction between a cloud user with its SaaS processes. Each process makes resource request to the cloud provider, who aggregates the resource requests and decides the resource allocation.

If the cloud provider knows the characteristics of its user processes, it might be able to manage resources in a centralized manner without resource requests. However, each SaaS process might have private information of the user and dynamic conditions that affect the resource requirements, e.g., instantaneous bandwidth capacity. Although, we assume a resource request/allocation communication paradigm, the real implementation of the resource requests can be conducted by SaaS users or automated for them by the SaaS provider.

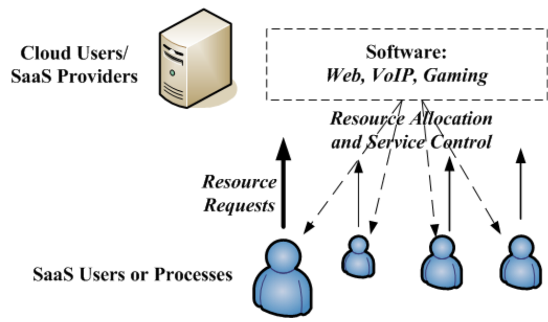


Fig. 3. Interaction between a cloud user and its SaaS processes.

B. Resource Allocation Mechanism

A cloud user decides the demand of VMs based on the arrival/departure of its SaaS processes. For the time being, we focus on a single VM and a fixed number of N SaaS processes running on the VM. We will consider the dynamic arrival/departure of SaaS processes in a later section. We assume that the VM has fixed amounts of divisible resources, e.g. CPU, bandwidth and RAM. Without loss of generality, we first focus on the allocation of a single resource with fixed capacity C. We will extend our model and results for multiple resources in a later section. We quantify the satisfaction of a SaaS process i by a valuation function $v_i(\cdot)$, where $v_i(d_i)$ defines the utility of process i when given d_i amount of the resource. We make the same assumption as in [12], [13] on the valuation function as follows.

Assumption *1:* For each i, over the domain $d_i \geq 0$ the valuation function $v_i(d_i)$ is concave, strictly increasing, and continuous; and over the domain $d_i > 0$, $v_i(d_i)$ is continuously differentiable. Furthermore, the right directional derivative at 0, denoted $v_i'(0)$, is finite.

Remark: The concavity assumption implies that although the more resource a process gets, the higher utility it values, the marginal utility diminishes when the allocation increases.

From a cloud user's perspective, a natural goal is to maximize the aggregate valuation of the SaaS processes, i.e., the *social welfare* of the system in the economics terminology. Mathematically, each cloud user tries to solve the following convex optimization problem:

$$\text{Maximize} \quad \sum_{i=1}^{N} v_i(d_i) \tag{1}$$

$$\text{Subject to} \quad \sum_{i=1}^{N} d_i \leq C; \tag{2}$$

$$d_i \geq 0, \quad i = 1, \cdots, N. \tag{3}$$

The above problem is not difficult to solve if the valuation functions are known to the cloud user; however, the challenge comes in when the valuation functions are private information of the SaaS users who do not want to reveal them voluntarily. Moveover, if requested to submit their valuations, SaaS users might misreport the true valuations of the processes so as to maximize their own allocation of resources and utilities.

To address the above problem, we use a resource bidding and allocation mechanism under which each SaaS process i submits a positive bid t_i and then obtains an amount of resource d_i. The resource allocation result $\mathbf{d} = \{d_1, \cdots, d_N\}$ is determined by a *proportional sharing* according to the bids:

$$d_i(\mathbf{t}) = \frac{t_i}{\sum_{j=1}^{N} t_j} C, \quad i = 1, \cdots, N, \tag{4}$$

where $\mathbf{t} = \{t_1, \cdots, t_N\}$ is the *bidding profile* of the processes. Throughout this paper, we use the notation d_i and $\mathbf{d}$ for two purposes. When expressed without parenthesis, they are pre-determined resource allocation result or requirement; when followed by parenthesis, i.e., $d_i(\cdot)$ and $\mathbf{d}(\cdot)$, they represent the proportional allocation function defined by Equation (4).

To achieve the above resource allocation, the SaaS providers only need to forward the bidding profile $\mathbf{t}$ to the cloud provider. The latter then uses each t_i as the weight for proportional scheduling, e.g. *weighted round robin* for CPU cycles and *weighted fair queueing* for bandwidth resource.

By bidding t_i, each SaaS process i will be penalized by $p_i \times t_i$ where p_i is the unit penalty for her bid. We denote u_i as the *utility function* of i defined by

$$u_i(\mathbf{t}, \mathbf{p}) = v_i(d_i(\mathbf{t})) - p_i t_i, \tag{5}$$

where $\mathbf{p} = \{p_1, \cdots, p_N\}$ is the *penalty profile* decided by the SaaS provider. Without a penalty mechanism, processes will bid as high as possible to obtain a large proportion of the resource. We introduce the penalty $\mathbf{p}$ as a control mechanism for the SaaS provider to reveal the underlying valuations of the processes and balance the resource allocation among them. Notice that the penalties do not have to be monetary, and can be any inconvenience imposed by the SaaS provider. We do not restrict the form of penalty in practice; however, we do assume that $p_i \times t_i$ will be the disutility to process i, measured in the same unit as the valuation, but not necessarily induces monetary benefits for the SaaS provider. When $\mathbf{p} = \mathbf{1}$, our mechanism coincides with the Kelly mechanism [14], under which the processes are penalized by how much they bid. We denote $\mathscr{M}^{\mathbf{p}}$ as the mechanism associated with the penalty vector $\mathbf{p}$. In particular, $\mathscr{M}^{\mathbf{1}}$ denotes the Kelly mechanism.

The interaction between the SaaS providers and users described in Figure 3 can be substantiated by the following three steps: 1) the SaaS provider will inform each SaaS process i its unit penalty p_i, 2) each process i submits its bid t_i to the provider, and 3) the SaaS provider broadcasts the *virtual price*

$$\mu = \frac{\sum_{j=1}^{N} t_j}{C}, \tag{6}$$

from which each process i can derive the amount of allocated resource by $d_i = t_i/\mu$. Despite of the differentiated penalties, our mechanism inherits the implementation simplicity of the Kelly mechanism in two ways: 1) the strategy space is still one-dimensional, i.e. a single bid; and 2) only a single feedback, i.e. μ, is required to send to all processes.

III. Resource Competition Game and Properties

In this section, we study the resource competition game induced by our resource allocation mechanism. The goal is to understand how the penalties affect the bidding behavior and the resulting resource allocation. In the next section, we will use the penalties to manage the resources for SaaS process.

A. Resource Competition Game

Given a resource allocation mechanism $\mathscr{M}^{\mathbf{p}}$, each SaaS process i tries to maximize its utility u_i. Kelly's original work [14] studies the *competitive equilibrium* of $\mathscr{M}^{\mathbf{1}}$, which is a bidding profile $\mathbf{t}$ that satisfies the following condition:

$$u_i(\mathbf{t}) = v_i(\frac{t_i}{\mu}) - t_i \geq v_i(\frac{\hat{t}_i}{\mu}) - \hat{t}_i, \ \forall \hat{t}_i \geq 0, \ i = 1, \cdots, N, \tag{7}$$

where μ is defined by Equation (6). The rationale of the competitive equilibrium relies on a *price-taking* assumption: the so-called *market clearing* price μ will not be affected by the action of a single player. Kelly proved that the resource allocation $\mathbf{d}(\mathbf{t})$ induced by the competitive equilibrium $\mathbf{t}$ solves the optimization problem defined in (1)-(3).

The price-taking assumption works well when the system has a large number of players such that a single bid's impact on the aggregation is infinitesimal. However, with only a few players at presence, each player's bid will have a large impact on μ as well as other players' resource allocation. Taking that into consideration, a further step is to model price-anticipating players and regard the resource allocation mechanism as a competition game, through which each player i chooses her strategy t_i to maximize her utility of u_i. More precisely, given a mechanism $\mathscr{M}^{\mathbf{p}}$, each i chooses the strategy t_i that solves:

$$\text{Maximize } u_i(t_i; \mathbf{t}_{-i}, \mathbf{p}) = v_i(d_i(\mathbf{t})) - p_i t_i, \tag{8}$$

where $\mathbf{t}_{-i}$ denotes the strategy profile of the players other than i. A strategy profile $\mathbf{t}^*$ is a *Nash equilibrium* of the resource competition game if for any i, the following is satisfied:

$$u_i(t_i^*; \mathbf{t}^*{}_{-i}, \mathbf{p}) \geq u_i(\hat{t}_i; \mathbf{t}^*{}_{-i}, \mathbf{p}), \quad \forall \hat{t}_i \geq 0. \tag{9}$$

Hajek and Gopalakrishnan [11] showed that the Kelly mechanism, i.e., $\mathscr{M}^{\mathbf{1}}$, induces a unique Nash equilibrium. Johari and Tsitsiklis [13] showed that the worst efficiency loss of this Nash equilibrium relative to the optimality of (1)-(3) is 25%. Despite of the well-studied Kelly mechanism, our generalization has not been studied yet. In the rest of this section, we will study the Nash equilibrium $\mathbf{t}^*$ of the generalized mechanism $\mathscr{M}^{\mathbf{p}}$ and the relationship between $\mathbf{p}$ and the resulting allocation $\mathbf{d}(\mathbf{t}^*)$.

B. Properties of $\mathscr{M}^{\mathbf{p}}$

We first show that for any strictly positive vector $\mathbf{p}$, the generalized mechanism $\mathscr{M}^{\mathbf{p}}$ induces a unique Nash equilibrium. The result is parallel to Theorem 2.2 of Johari [12], originated from Hajek and Gopalakrishnan [11]. We first define the feasible set of penalty vectors $\mathcal{P}$ to be strictly positive:

$$\mathcal{P} = \{\mathbf{p} | \ p_i > 0, \quad \forall i = 1, \cdots, N\}.$$

Remark: With $p_i \leq 0$, a process can keep increasing t_i to increase utility, and therefore no Nash equilibrium exists. We present the proofs of selected theorems in appendix.

Theorem *1 (Unique Nash equilibrium):* Suppose $N > 1$ and Assumption 1 holds. For any $\mathbf{p} \in \mathcal{P}$, there exists a unique Nash equilibrium $\mathbf{t} \geq 0$ for the resource competition game under $\mathscr{M}^{\mathbf{p}}$, and at least two components of $\mathbf{t}$ are positive.

Theorem 1 states that for any $\mathbf{p} \in \mathcal{P}$, there is a corresponding Nash equilibrium. Thus, we denote $\mathbf{t}^{\mathbf{p}}$ as the unique Nash equilibrium of $\mathscr{M}^{\mathbf{p}}$ that satisfies:

$$u_i(t_i^{\mathbf{p}}; \mathbf{t}^{\mathbf{p}}{}_{-i}, \mathbf{p}) \geq u_i(\hat{t}_i; \mathbf{t}^{\mathbf{p}}{}_{-i}, \mathbf{p}), \quad \forall \hat{t}_i \geq 0. \tag{10}$$

After knowing that each $\mathscr{M}^{\mathbf{p}}$ has a unique equilibrium, we ask the reverse question: Whether any resource allocation $\mathbf{d}$ can always be realized as a Nash equilibrium of some mechanism $\mathscr{M}^{\mathbf{p}}$? If the answer is positive, then the SaaS provider might be able to use $\mathbf{p}$ as a control mechanism to achieve its desirable resource allocation $\mathbf{d}$ for the system. Theorem 1 tells that at least two components of an equilibrium $\mathbf{t}$ are positive. This implies that the resulting allocation under our generalized mechanism is *non-dictatorial*, meaning no single process can obtain the whole capacity C. We denote $\mathcal{D}$ as the set of feasible non-dictatorial allocation defined as:

$$\mathcal{D} = \{\mathbf{d} \mid \sum_{j=1}^{N} d_j = C, \text{ and } \quad 0 \leq d_j < C, \ \forall j = 1, \cdots, N\}.$$

Next, we show that any non-dictatorial allocation can be implemented as a Nash equilibrium of some mechanism $\mathscr{M}^{\mathbf{p}}$.

Theorem *2 (Nash Implementation):* For any $\mathbf{d} \in \mathcal{D}$, there exists a mechanism $\mathscr{M}^{\mathbf{p}}$, whose unique Nash equilibrium $\mathbf{t}^{\mathbf{p}}$ induces the resource allocation of $\mathbf{d}$, i.e.

$$d_i(\mathbf{t}^{\mathbf{p}}) = d_i, \qquad i = 1, \cdots, N.$$

In particular, if $\mathbf{p}$ is defined by

$$p_i = h_i(d_i) = v_i'(d_i)(1 - \frac{d_i}{C}), \qquad i = 1, \cdots, N, \tag{11}$$

the Nash equilibrium strategy profile $\mathbf{t}^{\mathbf{p}}$ equals $\mathbf{d}$ as well.

Theorem 2 tells that with an appropriate penalty $\mathbf{p}$, any non-dictatorial allocation can be realized as the unique equilibrium of the resource competition game. However, the solution of Equation (11) depends on the valuation functions that are unknown to the SaaS provider. In the next section, we will characterize the desirable $\mathbf{p}$ and reach it by a feedback control algorithm.

Next, we explore the dynamics of penalty and the resulting allocation, i.e., how the resource allocation changes when the penalty vector varies. Intuitively, the resource allocated to a process should be monotonic in its unit penalty. In other words, by increasing the unit penalty of a particular process and keeping all the remaining fixed, this process's allocation should be non-increasing in equilibrium.

Theorem *3 (Monotonicity):* For any $\mathbf{p} \in \mathcal{P}$ and any process k, let a new vector $\mathbf{p}' \in \mathcal{P}$ be defined as:

$$p_i' \begin{cases} = p_i, & \text{if } i \neq k; \\ \neq p_i, & \text{if } i = k. \end{cases}$$

Let $d_i = d_i(\mathbf{t}^{\mathbf{p}})$ and $d_i' = d_i(\mathbf{t}^{\mathbf{p}'})$ be the resource allocation under $\mathscr{M}^{\mathbf{p}}$ and $\mathscr{M}^{\mathbf{p}'}$. Let $\rho = \sum_{i=1}^{N} t_i^{\mathbf{p}}/C$, $\rho' = \sum_{i=1}^{N} t_i^{\mathbf{p}'}/C$, and $h_i(d_i) = v_i'(d_i)(1 - d_i/C)$ for all i.

1) If $d_k > 0$ and $p_k' > p_k$, the resource allocation under $\mathscr{M}^{\mathbf{p}'}$ satisfies:

$$d_i' \begin{cases} < d_i, & \text{if } i = k; \\ > d_i, & \text{if } i \neq k \text{ and } p_i < h_i(d_i)/\rho'; \\ = d_i = 0, & \text{otherwise.} \end{cases}$$

2) If $d_k > 0$ and $p_k' < p_k$ or if $d_k = 0$ and $p_k' < h_k(0)/\rho$, the resource allocation under $\mathscr{M}^{\mathbf{p}'}$ satisfies:

$$d_i' \begin{cases} > d_i, & \text{if } i = k; \\ < d_i, & \text{if } i \neq k \text{ and } d_i > 0; \\ = d_i, & \text{if } i \neq k \text{ and } d_i = 0. \end{cases}$$

3) If $d_k = 0$ and $p_k' \geq h_k(0)/\rho$, the resource allocation under $\mathscr{M}^{\mathbf{p}'}$ satisfies:

$$d_i' = d_i, \quad \forall i = 1, \cdots, n.$$

Theorem 3 identifies the conditions under which the monotonicity properties holds. Case 1 and 2 says when a process had a positive amount of resource, after increasing or decreasing its unit penalty, the amount of resource is decreased or increased respectively. However, case 2 and 3 identifies a penalty threshold $h_k(0)/\rho$, below which a zero-resource process will start to obtain a positive amount of resource. In practice, the monotonicity property enables the SaaS provider to dynamically and gracefully adjust the penalty vector so as to control and achieve desirable resource allocations.

Although each $\mathscr{M}^{\mathbf{p}}$ induces a unique Nash equilibrium $\mathbf{t}^{\mathbf{p}}$, two different mechanisms might induce the same resource allocation and process utilities. We define the equivalence class of mechanisms as follows.

Definition *1:* We define two mechanisms $\mathscr{M}^{\mathbf{p}}$ and $\mathscr{M}^{\mathbf{q}}$ to be *equivalent*, if they induce the same resource allocation and process utilities in equilibrium, i.e. $\mathbf{d}(\mathbf{t}^{\mathbf{p}}) = \mathbf{d}(\mathbf{t}^{\mathbf{q}})$ and $\mathbf{u}(\mathbf{t}^{\mathbf{p}}, \mathbf{p}) = \mathbf{u}(\mathbf{t}^{\mathbf{q}}, \mathbf{q})$ for all $i = 1, \cdots, N$.

For any $\mathbf{p} \in \mathcal{P}$, we denote $\mathcal{E}^{\mathbf{p}}$ as the set of processes that get zero allocation in equilibrium, defined as the following:

$$\mathcal{E}^{\mathbf{p}} = \{i | t_i^{\mathbf{p}} = 0\}. \tag{12}$$

Based on the result of Theorem 3, we can characterize a class of equivalent mechanisms as the following.

Corollary *1 (Zero-allocation Equivalence):* For any $\mathbf{p} \in \mathcal{P}$, let the set $\mathcal{Q}^{\mathbf{p}}$ be

$$\mathcal{Q}^{\mathbf{p}} = \{\mathbf{q} \mid q_i = p_i, \quad \forall i \notin \mathcal{E}^{\mathbf{p}}; \quad q_i \geq p_i, \quad \forall i \in \mathcal{E}^{\mathbf{p}}\}.$$

Then for any $\mathbf{q} \in \mathcal{Q}^{\mathbf{p}}$, the unique Nash equilibrium of $\mathscr{M}^{\mathbf{q}}$ is $\mathbf{t}^{\mathbf{q}} = \mathbf{t}^{\mathbf{p}}$. Moreover, $\mathscr{M}^{\mathbf{q}}$ is equivalent to $\mathscr{M}^{\mathbf{p}}$.

Corollary 1 tells that for the processes that get zero resource in an equilibrium, keep increasing their unit penalties will not change the equilibrium. Processes will use the same strategy profile and keep the same penalties and utilities.

Theorem *4 (Linear Equivalence):* For any $\mathbf{p}, \mathbf{q} \in \mathcal{P}$ with $\mathbf{q} = k\mathbf{p}$ for some $k > 0$, the unique Nash equilibrium of $\mathscr{M}^{\mathbf{q}}$ is $\mathbf{t}^{\mathbf{q}} = \frac{1}{k}\mathbf{t}^{\mathbf{p}}$. Moreover, $\mathscr{M}^{\mathbf{q}}$ is equivalent to $\mathscr{M}^{\mathbf{p}}$.

Theorem 4 states that when the penalty vector is scaled by a positive constant k, the resource allocation does not change in equilibrium. However, the strategy profile scales $1/k$ and keeps the penalties unchanged. As a result, the utility of the processes will be the same as well in both equilibria. A simple consequence of this theorem is the following.

Corollary *2 (Equivalent Kelly Mechanisms):* For any scalar $k > 0$, the mechanism $\mathscr{M}^{\mathbf{k}}$ is equivalent to the Kelly mechanism $\mathscr{M}^{\mathbf{1}}$.

Theorem 1 implies that any $\mathscr{M}^{\mathbf{p}}$ maps to a unique resource allocation $\mathbf{d}$. Theorem 4 and Corollary 1 imply that this mapping can be many-to-one. Theorem 2 states that the mapping from $\mathscr{M}^{\mathbf{p}}$ to $\mathcal{D}$ is indeed onto. From Theorem 4, we know that the mechanisms that have linearly dependent $\mathbf{p}$ vectors are equivalent. Without loss of generality, we focus on the domain $\hat{\mathcal{P}} \subseteq \mathcal{P}$ defined inside a unit simplex as:

$$\hat{\mathcal{P}} = \{\mathbf{p} \mid \sum_{j=1}^{N} p_j = 1 \quad and \quad p_j > 0, \quad j = 1, \cdots, N\}.$$

The following theorem reveals a one-to-one and onto mapping relationship between $\mathcal{D}$ and a subset of $\hat{\mathcal{P}}$.

Theorem *5 (Mapping):* There exists a connected set $\tilde{\mathcal{P}} \subseteq \hat{\mathcal{P}}$ such that, the mapping $f : \tilde{\mathcal{P}} \to \mathcal{D}$ defined by

$$f(\mathbf{p}) = \mathbf{d}(\mathbf{t}^{\mathbf{p}}), \qquad \forall \mathbf{p} \in \tilde{\mathcal{P}}$$

is continuous and bijective. Particularly, if $n = 2$, then $\tilde{\mathcal{P}} = \hat{\mathcal{P}}$.

If we focus on the set $\hat{\mathcal{P}}$, we eliminates the linear dependency of the $\mathbf{p}$ vectors addressed in Theorem 4. Theorem 5 states that there always exists a subset of $\tilde{\mathcal{P}} \subseteq \hat{\mathcal{P}}$ that maps to $\mathcal{D}$ continuously and bijectively. The mapping from $\hat{\mathcal{P}}$ to $\mathcal{D}$ might still be many-to-one, because there might still exist equivalent mechanisms addressed in Corollary 1. However, when $n = 2$, $\hat{\mathcal{P}}$ maps to $\mathcal{D}$ bijectively, because no process gets zero allocation in equilibrium.

C. Interpretation and Illustration

In this subsection, we connect the properties of our mechanism into a big picture and use two examples to illustrate.

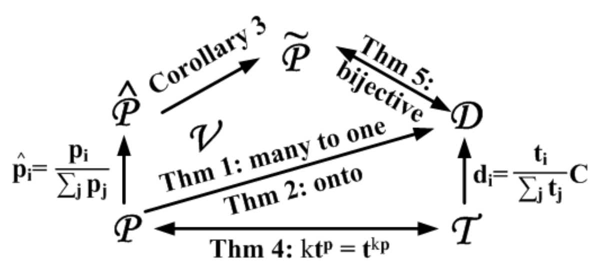


Fig. 4. Mapping between penalty and allocation.

Figure 4 visualizes the relationship between the domain of mechanisms $\mathscr{M}^{\mathbf{p}}$ and the resulting resource allocations $\mathbf{d}$. $\mathcal{T}$ defines the space of feasible bidding profiles. $\mathcal{V}$ defines the space of valuation functions under Assumption 1. By Theorem 1 and 2, we know that each $\mathbf{p}$ maps onto $\mathcal{D}$; however, the mapping is many-to-one due to the linearly equivalent mechanisms addressed in Theorem 4. If we normalize every $\mathbf{p} \in \mathcal{P}$ into $\hat{\mathcal{P}}$, the mapping from $\hat{\mathcal{P}}$ to $\mathcal{D}$ might still be many-to-one. After reducing the equivalent mechanisms addressed in Corollary 1, we finally obtain $\tilde{\mathcal{P}}$ that maps to $\mathcal{D}$ bijectively. The exact set of $\tilde{\mathcal{P}}$ and the mapping to $\mathcal{D}$ totally depend on the underlying valuation space $\mathcal{V}$ of the processes.

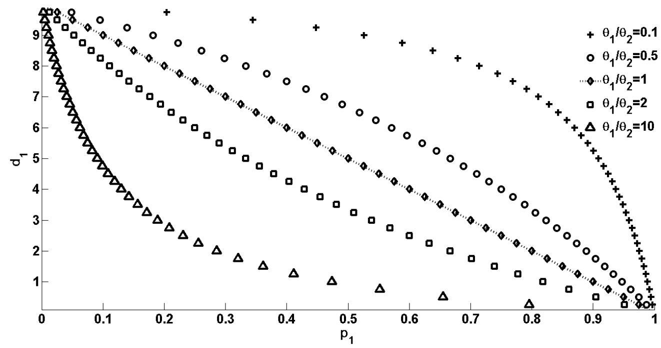


Fig. 5. An example of 2 SaaS processes with linear valuation.

In the first example, we have $N = 2$, $C = 10$, $v_1(d_1) = \theta_1 d_1$ and $v_2(d_2) = \theta_2 d_2$ for some positive constant θ_1 and θ_2. Figure 5 illustrates the resource allocation for both processes under various $\mathbf{p} \in \hat{\mathcal{P}}$. p_1 varies along x-axis and $1 - p_1$ corresponds to p_2. On the y-axis, we plot $d_1(\mathbf{p})$, i.e. the resource allocation to process 1 in equilibrium. We can also easily identify the resource allocation to process 2 as $C - d_1(\mathbf{p})$ correspondingly in the figure. We observe that the resource allocation depends on both the penalty and the ratio of θ_1/θ_2. Particularly, when $\theta_1 = \theta_2$, the resource allocation is inversely proportional to the penalty of the processes.

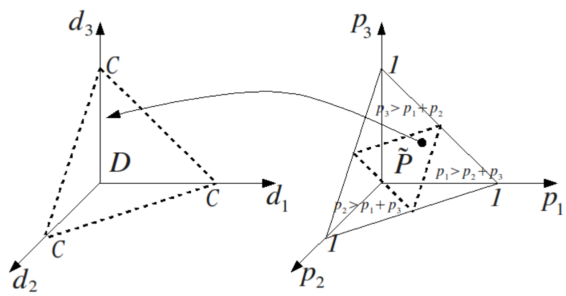


Fig. 6. An example of 3 processes with linear valuation.

In the second example, we have $N = 3$, and $v_1(x) = v_2(x) = v_3(x) = \theta x$ for some $\theta > 0$. Figure 6 illustrates the mapping from the penalty simplex $\hat{\mathcal{P}}$ to the non-dictatorial resource allocation set $\mathcal{D}$. In particular, there is a proper subset $\tilde{\mathcal{P}} = \{\mathbf{p} \mid 0 < p_i \le \frac{1}{2}, \quad i = 1, 2, 3\}$ that maps to $\mathcal{D}$ bijectively. The points in $\hat{\mathcal{P}} \backslash \tilde{\mathcal{P}}$ are mapped to the triangular boundary of $\mathcal{D}$, where one of processes gets zero allocation. Readers can also check that the mapping from $\tilde{\mathcal{P}}$ to $\mathcal{D}$ satisfies

$$d_i(\mathbf{p}) = (1 - 2p_i)C, \quad i = 1, 2, 3.$$

IV. Revealing Valuations and Optimizing Efficiency

By Theorem 2, any non-dictatorial resource allocation can be achieved as a Nash equilibrium. Thus, in theory, we can close the 25% efficiency gap by choosing an appropriate penalty $\mathbf{p}$ that maximizes the social welfare. In practice, however, we need to know the valuation functions that are private information and may not be disclosed by the SaaS users. Next, we try to derive the hidden valuations via observing the resulting bidding profile of the processes as follows.

Theorem *6 (Observability of Marginal Utility):* Suppose for any $\mathbf{p} > 0$, we can observe the bids $\mathbf{t}^{\mathbf{p}}$. For any non-zero bid $t_i^{\mathbf{p}}$, let $d_i = d_i(\mathbf{t}^{\mathbf{p}})$ be the equilibrium allocation for i. Then the marginal utility of process i at the resource level of d_i is

$$v_i'(d_i) = \frac{p_i}{C - d_i} \sum_{j=1}^{N} t_j^{\mathbf{p}}. \tag{13}$$

Theorem 6 states that by observing the bidding profile, we can derive processes' marginal utilities at the allocated resource level. By the monotonicity property of Theorem 3, we can gradually change one process's penalty to achieve various resource levels for that process. Thus, by calculating the marginal utility of a process at various resource level, we can fully reveal its hidden valuation function in principle. However, this method might be tedious and slow. To maximize the social welfare, we explore the direct relationship between the optimal resource allocation and the corresponding penalty.

Theorem *7 (Condition of Optimality):* Suppose the optimality of problem (1)-(3) is a strictly positive resource allocation vector, i.e. $\mathbf{d}^* \in \mathcal{D}$ and $d_i^* > 0$ for all $i = 1, \cdots, N$. A vector $\mathbf{p}^* > 0$ induces the unique Nash equilibrium with the allocation $\mathbf{d}^*$ if and only if the following condition is satisfied:

$$p_i^* : p_j^* = C - d_i^* : C - d_j^*, \quad \forall i, j = 1, \cdots, N. \tag{14}$$

In particular, when $N = 2$, $\mathbf{d}^*$ is achieved when both processes incur the same amount of penalty, i.e. $p_1^* t_1^{\mathbf{P}} = p_2^* t_2^{\mathbf{P}}$.

Theorem 7 states that for any pair of processes i and j, the optimal penalty ratio should equal the ratio of $C - d_i^* : C - d_j^*$. This result not only gives us a way to verify the optimality without knowing hidden valuations, but it also provides a heuristic feedback control mechanism to update the penalty vector so as to achieve the optimality.

A Feedback Control Mechanism: We update the penalty vector $\mathbf{p}$ at the beginning of each time period and assume that the length is long enough for the processes to reach the Nash equilibrium by the end of the time period. We start with the Kelly mechanism (by setting all p_is to be the same) and reach a resource allocation $\mathbf{d}[0]$ by the end of time period 0. If a process i joins in at time k, $p_i[k]$ will be set to be the average price in the system. Otherwise, at the beginning of each time period k, we update the $\mathbf{p}$ vector as the following:

$$p_i[k] = C - d_i[k-1] \quad \forall i = 1, \cdots, N.$$

To illustrate, let us consider the following experiments. We have five SaaS processes, each are CPU-bound. The valuation of each of these five processes are depicted in Figure 7. In particular, process 2 and 5 have a linear valuation function while other processes have different concave valuation functions.

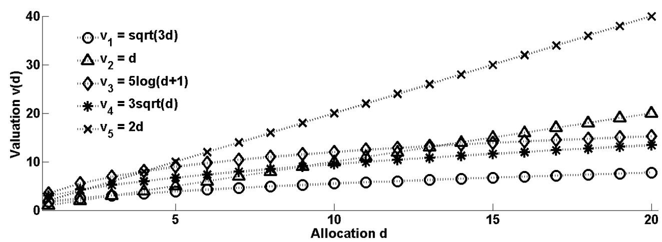
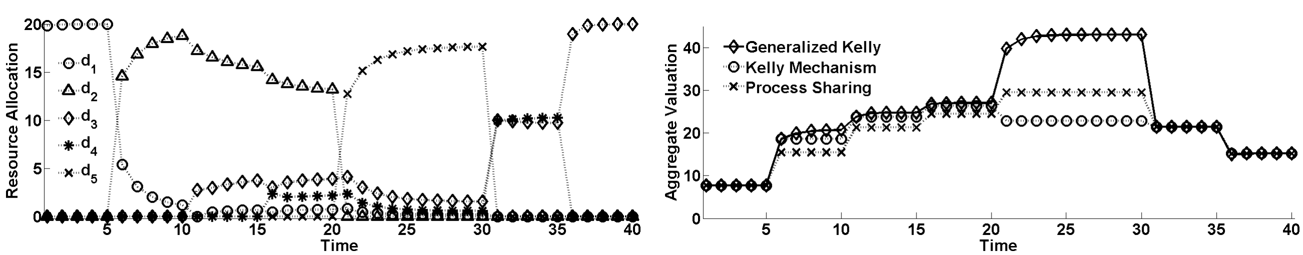


Fig. 7. An example of 5 SaaS processes with different valuation functions.

We perform two experiments wherein SaaS processes arrive and depart at different times. For experiment A, the five processes stay in the system during the time interval (1,30), (6,20), (11,40), (16,35) and (21,30) respectively. For experiment B, the five processes stay in the system during the time interval (1,40), (21,30), (21,40), (1,35) and (11,25) respectively. Figure 8(a) and 9(a) show instantaneous resource allocation to each process while Figure 8(b) and 9(b) show the instantaneous aggregate valuation in both experiments respectively. One can observe that our resource allocation and feedback mechanism can (a) dynamically adapt to the changing resource demand of SaaS processes; (b) quickly converge to the optimal resource allocation level; (c) allocate more resource to linear valuation function, where the efficiency gap is big for the Kelly mechanism.



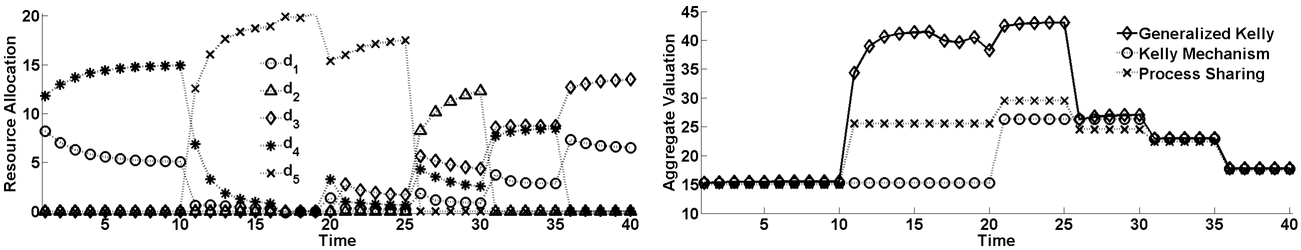
(a) Resource Allocation (b) Social Welfare

Fig. 8. Experiment A.

V. Multiple Resources

In this section, we generalize our results to a multi-resource context, where SaaS processes need heterogeneous resources, e.g. CPU and network bandwidth, at the same time. We consider R divisible resources. Each resource r has a capacity of C_r. We define a $R \times N$ resource allocation matrix D as

$$\mathbf{D} = (\mathbf{d_1}, \cdots, \mathbf{d_N}) = \begin{pmatrix} d_{11} & \dots & d_{1N} \\ \vdots & \ddots & \vdots \\ d_{R1} & \dots & d_{RN} \end{pmatrix} = \begin{pmatrix} D_1 \\ \vdots \\ D_R \end{pmatrix}.$$



(a) Resource Allocation (b) Social Welfare

Fig. 9. Experiment B

Each $\mathbf{d_i}$ is a column vector that represents the resource allocation to process i and each D_r denotes a row vector that represents the distribution of resource r to N processes.

Each process i's valuation is defined by $v_i(\mathbf{d_i})$. We generalize the assumption on the valuation function as the following.

Assumption *2:* For each i, over the domain $\mathbf{d_i} \geq \mathbf{0}$ the valuation function $v_i(d_i)$ is quasi-concave, strictly increasing, continuous, and continuously differentiable. Furthermore, the right directional partial derivative for any resource r at 0, denoted $\partial_r v_i(0)$, are finite.

Remark: Quasi-concavity, a weaker condition than concavity, requires the upper contour sets $\{\mathbf{d_i} | v_i(\mathbf{d_i}) \geq v\}$ be convex. Intuitively, it means that if two bundles of resources are valued higher the current bundle, then a linear mixture of these two bundle values higher than the current bundle as well.

The social welfare maximization problem becomes:

$$\text{Maximize } \sum_{i=1}^{N} v_i(\mathbf{d_i}) \tag{15}$$

$$\text{Subject to } \quad \sum_{i=1}^{N} d_{ri} \leq C_r, \quad r = 1, \cdots, R; \tag{16}$$

$$d_{ri} \geq 0, \quad r = 1, \cdots, R, \; i = 1, \cdots, N. \tag{17}$$

Similar to $\mathbf{D}$, we define matrices $\mathbf{P} = \{p_{ri}\}$ and $\mathbf{T} = \{t_{ri}\}$. p_{ri} denotes the unit penalty for i to bid resource r and t_{ri} denotes its bid. We denote $\mathbf{p_i}$ and $\mathbf{t_i}$ as the penalty and bid with respect to process i and P_r and T_r as the penalty and bid with respect to resource r. Each r is allocated by:

$$d_{ri}(T_r) = \frac{t_{ri}}{\sum_{j=1}^{N} t_{tj}} C_r, \quad i = 1, \cdots, N. \tag{18}$$

We denote $\mathscr{M}^{\mathbf{P}}$ as the joint allocation mechanism for R resources. Accordingly, process i obtains a utility of

$$u_i(\mathbf{T}, \mathbf{P}) = v_i(\mathbf{d_i}(\mathbf{T})) - \mathbf{p_i}^{\mathsf{T}} \mathbf{t_i}. \tag{19}$$

A bidding profile matrix $\mathbf{T}^*$ is a *Nash equilibrium* if for any process i, the following is satisfied:

$$u_i(\mathbf{t_i^*}; \mathbf{t^*}_{-\mathbf{i}}, \mathbf{P}) \geq u_i(\mathbf{t_i}; \mathbf{t^*}_{-\mathbf{i}}, \mathbf{P}), \quad \forall \mathbf{t_i} \geq \mathbf{0}. \tag{20}$$

We focus on the non-degenerated cases where each resource has multiple processes to compete for. We define the domain of non-dictatorial allocation for resource r as follows:

$$\mathcal{D}_r = \{D_r \mid \sum_{i=1}^{N} d_{ri} = C_r, \; 0 \leq d_{ri} < C_r, \; \forall i = 1, \cdots, N\}.$$

Similarly, the domain of penalty for resource r is defined as:

$$\mathcal{P}_r = \{P_r \mid p_{ri} > 0, \; \forall i = 1, \cdots, N\}.$$

Then we redefine the domain of resource allocation $\mathcal{D}$ and penalty $\mathcal{P}$ for multiple resources using the Cartesian products of the individual resource domains as follows:

$$\mathcal{D} = \prod_{r=1,\cdots,R} \mathcal{D}_r, \qquad \mathcal{P} = \prod_{r=1,\cdots,R} \mathcal{P}_r.$$

Although the multi-resource context enlarges the strategy space, many desirable properties are inherited. We present the parallel results corresponding to Theorem 1, 2, 6 and 7.

Theorem *8 (Uniqueness of Nash equilibrium):* Suppose any $\mathbf{P} \in \mathcal{P}$, $N > 1$ and Assumption 2 holds. Then there exists a unique Nash equilibrium $\mathbf{T}$ for the resource competition game of mechanism $\mathscr{M}^{\mathbf{P}}$.

Theorem *9 (Nash Implementation):* For any $\mathbf{D} \in \mathcal{D}$, there exists a mechanism $\mathscr{M}^{\mathbf{P}}$, whose unique Nash equilibrium $\mathbf{T^P}$ will yield the resource allocation of $\mathbf{D}$, i.e.

$$\mathbf{d_i}(\mathbf{T^P}) = \mathbf{d_i}, \qquad i = 1, \cdots, N.$$

In particular, if $\mathbf{P}$ is defined by

$$p_{ri} = \partial_r v_i(\mathbf{d_i})(1 - \frac{d_{ri}}{C_r}), \quad r = 1, \cdots, R, \; i = 1, \cdots, N,$$

the Nash equilibrium bidding profile $\mathbf{T^P}$ equals $\mathbf{D}$ as well.

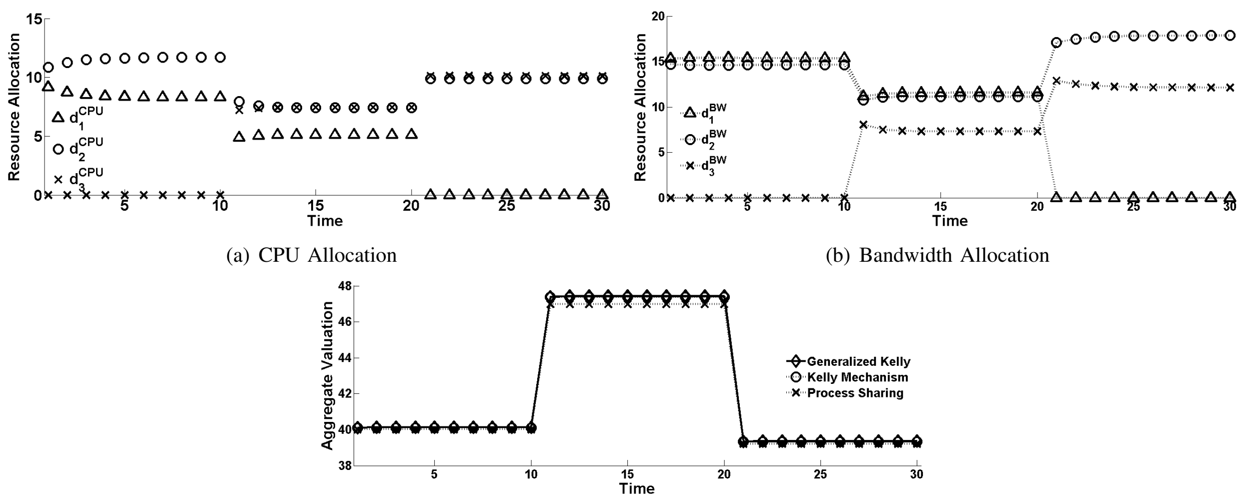
Theorem *10 (Observability of Marginal Utility):* Suppose for any $\mathbf{P} \in \mathcal{P}$, we can observe the equilibrium bids $\mathbf{T^P}$. Let $\mathbf{d_i} = \mathbf{d_i}(\mathbf{T^P})$ be the allocation for i. For any non-zero bid $t^{\mathbf{P}}_{ri}$, process i's marginal utility with respect to r is

$$\partial_r v_i(\mathbf{d_i}) = \frac{p_{ri}}{C_r - d_{ri}} \sum_{j=1}^{N} t^{\mathbf{P}}_{rj}. \tag{21}$$

Theorem *11 (Condition of Optimality):* Suppose the optimality to the problem (15)-(17) is a non-dictatorial resource allocation, i.e. $\mathbf{D}^* \in \mathcal{D}$ and $d^*_{ri} > 0$ for all $r = 1, \cdots, R$ and $i = 1, \cdots, N$. A matrix $\mathbf{P} \in \mathcal{P}$ will induce a unique Nash equilibrium with the same resource allocation $\mathbf{D}^*$ if and only if for every resource r the following condition is satisfied:

$$p^*_{ri} : p^*_{rj} = C_r - d^*_{ri} : C_r - d^*_{rj}, \quad \forall i, j = 1, \cdots, N. \tag{22}$$

We carry out two experiments and consider multi-resource allocation. We have three SaaS processes, each of these processes needs both the CPU and network bandwidth resources. Process 1, 2 and 3 stay in the system during the time interval (1,20), (1,30) and (11,30) respectively. The total CPU resource is $C_1 = 20$ and total bandwidth resource is $C_2 = 30$. The valuation function for three SaaS processes in Experiment C are: $v_1(x, y) = \log(18x+1)\log(2y+1)$, $v_2(x, y) = \log(9x+1)\log(9y+1)$ and $v_3(x, y) = \log(2x + 1)\log(18y + 1)$. For Experiment D, the valuation functions of these processes are: $v_1(x, y) = x\log(2y + 1)$, $v_2(x, y) = \log(9x + 1)\log(9y + 1)$ and $v_3(x, y) = y\log(2x + 1)$. Note that in both experiments, process 1 is more CPU intensive while process 3 is more bandwidth intensive. However, in Experiment D, the linear component of valuation functions imply that the CPU and bandwidth resources bring much higher valuations to process 1 and 3 compared to those in Experiment C.



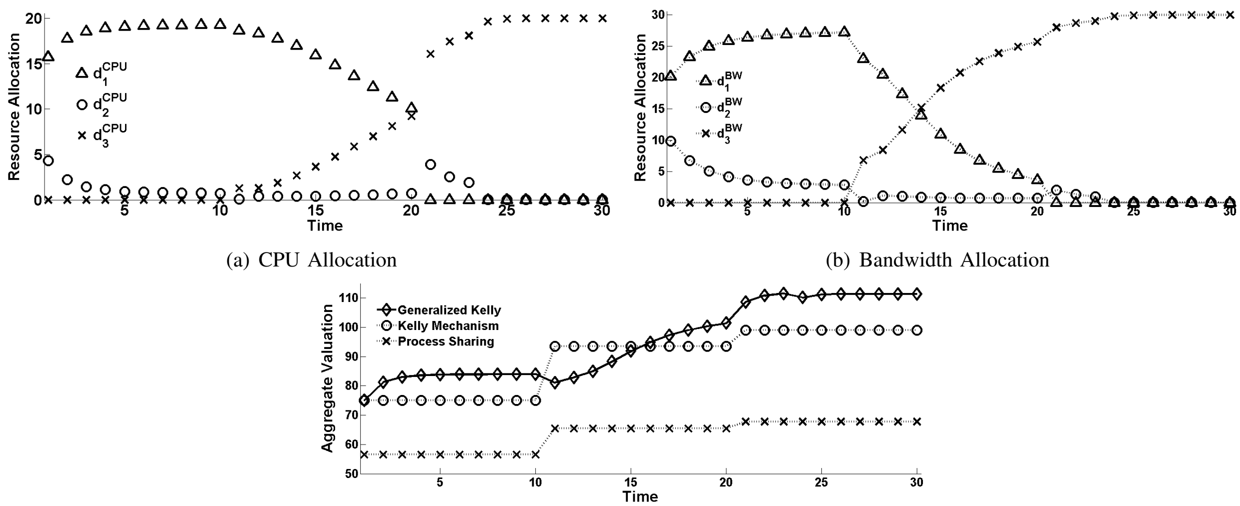
(a) CPU Allocation

(b) Bandwidth Allocation

(c) Social Welfare

Fig. 10. Experiment C

Figure 10 and 11 illustrate the resource allocations and the social welfare in both experiments respectively. We see that our algorithm can efficiently and adaptively allocate resources. In Experiment C, the mechanism balances the allocation so as to maximize the social welfare, and does not allocate plenty of CPU resource to process 1 (and does not allocate plenty of bandwidth resource to process 3 as well). In contrast, the mechanism allocates most of the CPU resource to process 1 (and most of the bandwidth resource to process 3) to maximize the social welfare in Experiment D, in which we can close a relatively big efficient gap of the Kelly mechanism.



(a) CPU Allocation

(b) Bandwidth Allocation

(c) Social Welfare

Fig. 11. Experiment D

VI. Related Work

Cloud computing has received increased attention within both the IT industry [1], [2], [3] and the research community [4], [5]. Due to the severe competition among cloud providers, the prices of cloud computing services keep declining; however, the commoditized services deemphasize service requirements [8], e.g. guaranteed levels of performance and availability. This work proposes a framework to support resource management and service controls for the SaaS applications.

Resource allocation [14], [12] and pricing [18], [9] for network resources, e.g. bandwidth, have been studied extensively during the last decade. Among various proposed pricing and allocation mechanisms, the Kelly mechanism [14] stands out as a simple and practical mechanism with low strategic flexibility (one-dimensional space) and low pricing flexibility (single price) [13]. It was argued in [13] that if we increase the strategic flexibility while preserving the single price restriction, the efficiency loss can be arbitrarily large.

To achieve efficiency, many mechanisms have been designed by introducing pricing flexibility into the proportional allocation mechanism. Maheswaran and Basar [15], [16] modified the proportional allocation rule by adding a parameter ϵ to the total bids in the denominator and designed explicit price functions for players with different valuation functions. Nguyen and Vojnovic [17] introduced weights to the proportional allocation function and studied the revenue maximization problem for the resource provider. In our mechanism, the proportional allocation remains the same; however, the players are penalized at different constant rates. Our objective is to maximize social welfare instead of the total penalties.

Another major line of extending pricing flexibility is to apply the celebrated Vickrey-Clarke-Groves (VCG) [20], [6], [10] mechanism with the proportional allocation rule. Yang and Hajek [21], [22] and Johari and Tsitsiklis [13] independently design VCG-type of mechanism with one-dimensional bids from the players. Dimakis, Jain and Walrand [7] considered VCG mechanism in multiple divisible goods allocation for a two-dimensional strategy space. Stoenescu and Ledyard [19] designed a mechanism that can implement efficient allocation in the sense of Pareto optimality as a Nash equilibrium in a two-dimensional strategy space. We introduce the pricing flexibility as a build-in parameter, integrated into the proportional allocation mechanism itself; therefore, we view our mechanism as a generalization of the Kelly mechanism instead of an add-on pricing mechanism to the players.

VII. Conclusions

In this paper, we propose a resource management framework for SaaS providers to allocate resources and control service levels of their SaaS processes. Under this framework, we allocate resources proportional to processes's request bids; however, processes are penalized by their bids as well. By differentiating the unit penalty of different processes, we generalized the Kelly mechanism and show that the competition game under the generalized mechanism satisfies desirable properties: 1) each mechanism under a diffent penalty vector has a unique Nash equlibrium, 2) any non-dictatorial resource allocation can be realized as the unique Nash equilibrium of a penalty vector. By observing the bidding profile in equilibrium, we can derive the underlying valuation functions of the SaaS processes. We further characterize the optimality condition of the penalty and design a feed-back control algorithm to maximize the social welfare. We show that our feed-back control mechanism adapts to dynamic process arrival/departure and closes the efficiency gap of the Kelly mechanism.

REFERENCES

[1] Amazon elastic compute cloud (ec2). *http://www.amazon.com/ec2*.

[2] Google app engine. *http://appengine.google.com*.

[3] Microsoft windows azure. *http://www.microsoft.com/windowsazure/*.

[4] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. Above the clouds: A berkeley view of cloud computing. *Technical Report EECS-2009-28, UC Berkeley*, 2009.

[5] R. Buyya, C. S. Yeo1, and S. Venugopal. Market-oriented cloud computing: Vision, hype, and reality of delivering computing as the 5th utility. *Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid*, page 1, 2009.

[6] E. H. Clarke. Multipart pricing of public goods. *Public Choice*, 11:19–33, 1971.

[7] A. Dimakis, R. Jain, and J. Walrand. Mechanisms for efficient allocation in divisible capacity networks. *In Proceedings of the 45th IEEE Conference on Decision and Control, San Diego, CA, USA*, December 2006.

[8] D. Durkee. Why cloud computing will never be free. *Communications of the ACM*, 53(5), May 2009.

[9] M. Falkner, M. Devetsikiotis, and I. Lambadaris. An overview of pricing concepts for broadband IP networks. *IEEE Communications Surverys*, 3(2), 2000.

[10] T. Groves. Incentives in teams. *Econometrica*, 41(4):617–631, 1973.

[11] B. Hajek and G. Gopalakrishnan. Do greedy autonomous systems make for a sensible Internet? *Presented at the Conference on Stochastic Networks, Stanford University*, 2002.

[12] R. Johari. Efficiency loss in market mechanisms for resource allocation. *PhD thesis, Massachusetts Institute of Technology*, June 2004.

[13] R. Johari and J. N. Tsitsiklis. Efficiency of scalar-parameterized mechanisms. *Operation Research*, 2008.

[14] F. P. Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, 8:33–37, 1998.

[15] R. T. Maheswaran and T. Basar. Nash equilibrium and decentralized negotiation in auctioning divisible resources. *Group Decision and Negotiation*, 12(5):361–395, 2003.

[16] R. T. Maheswaran and T. Basar. Efficient signal proportional allocation (ESPA) mechanisms: Decentralized social welfare maximization for divisible resources. *IEEE Journal on Selected Areas of Communications*, 24(5):1000–1009, 2006.

[17] T. Nguyen and M. Vojnovic. The weighted proportional allocation mechanism. *Microsoft Technical Report MSR-TR-2009-123*, September 2009.

[18] S. Shenker, D. Clark, D. Estrin, and S. Herzog. Pricing in computer networks: Reshaping the research agenda. *Telecommunications Policy*, 20(3):183–201, 1996.

[19] T. Stoenescu and J. O. Ledyard. Implementation in Nash equilibria of a rate allocation problem in networks. *Forty-fourth Annual Allerton Conference*, September 2006.

[20] W. Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16(1):8–37, 1961.

[21] S. Yang and B. Hajek. Revenue and stability of a mechanism for effocoent allocation of a divisible good. *Preprint*, 2005.

[22] S. Yang and B. Hajek. VCG-Kelly mechanisms for allocation of divisible goods: Adapting VCG mechanisms to one-dimensional signals. *IEEE Journal on Selected Areas of Communications*, 25(6):1237–1243, 2007.

APPENDIX

Lemma *1 (Nash equilibrium condition):* A strategy profile $\mathbf{t}$ is a Nash equilibrium of the mechanism $\mathscr{M}^{\mathbf{p}}$ if and only if at least two components of $\mathbf{t}$ are positive, and for each i, the following conditions hold:

$$\frac{1}{p_i} v_i'\Big(\frac{t_i C}{\sum_{j=1}^N t_j}\Big)\Big(1 - \frac{t_i}{\sum_{j=1}^N t_j}\Big) = \frac{\sum_{j=1}^N t_j}{C}, \text{ if } t_i > 0; \tag{23}$$

$$\frac{1}{p_i} v_i'(0) \le \frac{\sum_{j=1}^N t_j}{C}, \qquad \text{if } t_i = 0. \tag{24}$$

Proof of Lemma 1: The proof follows the same argument of Hajek and Gopalakrishnan [11] and Johari [12] and takes the price vector $\mathbf{p}$ as a new parameter.

Step 1: *If* $\mathbf{t}$ *is a Nash equilibrium, at least two components of* $\mathbf{t}$ *are positive.* Suppose we have a strategy profile $\mathbf{t}$ such that only $t_i > 0$ for some user i and $t_j = 0$ for all $j \neq i$. This strategy profile cannot be an equilibrium, because user i can always be better off by reducing the bid t_i slightly. However, $\mathbf{t} = \mathbf{0}$ cannot be an equilibrium as well, because any user can be better off by bidding an infinitesimal amount and obtains the whole bandwidth. Thus, in equilibrium, $\mathbf{t}$ must have at least two positive components.

Step 2: *For any* $\mathbf{p} > 0$ *and* $\mathbf{t} \geq 0$ *with at least two positive components, the function* $u_i(t_i; \mathbf{t}_{-i}, \mathbf{p})$ *is strictly concave and continuously differentiable in* t_i*, for* $t_i \geq 0$. Because $\mathbf{t}$ has at least two positive components, the utility function u_i can be written as

$$u_i(t_i; \mathbf{t}_{-i}, \mathbf{p}) = v_i(\frac{t_i}{t_i + \sum_{j \neq i} t_j} C) - p_i t_i.$$

Because $t_i/(t_i + \sum_{j\neq i} t_j)$ is a strictly increasing function of t_i (for $t_i \geq 0$) and $v_i(\cdot)$ is a strictly increasing, concave, and differentiable function by assumption, by extracting a linear function $p_i t_i$, u_i is also a strictly increasing, concave, and differentiable function in t_i.

Step 3: Let $\mathbf{t}$ be a Nash equilibrium. By Steps 1 and 2, $\mathbf{t}$ has at least two positive components and $u_i(t_i; \mathbf{t}_{-i}, \mathbf{p})$ is strictly concave and continuously differentiable in $t_i \geq 0$. Thus t_i must be the unique maximizer of $u_i(t_i; \mathbf{t}_{-i}, \mathbf{p})$ over $t_i \geq 0$, and satisfy the following first order optimality conditions:

$$\frac{\partial u_i}{\partial t_i}(t_i; \mathbf{t}_{-i}, \mathbf{p}) \begin{cases} = 0, & \text{if } t_i > 0; \\ \leq 0, & \text{if } t_i = 0. \end{cases}$$

By multiplying $\sum_{j=1}^{N} t_j/C$, the above conditions become the conditions (23)-(24).

Conversely, if we have a strategy profile $\mathbf{t}$ with at least two positive components, by Step 2, we know $u_i(t_i;\mathbf{t}_{-i},\mathbf{p})$ is strictly concave and continuously differentiable in $t_i \geq 0$. The conditions (23)-(24) imply that t_i maximizes $u_i(t_i;\mathbf{t}_{-i},\mathbf{p})$ over $t_i \geq 0$. Thus $\mathbf{t}$ is a Nash equilibrium.

∎

Proof of Theorem 1: The proof uses Lemma 1 and follows the same argument of Johari [12].

Step 1: *The function $\hat{v}_i$ defined in (**??**) is strictly concave and strictly increasing over $0 \leq d_i \leq C$.* By differentiating $\hat{v}_i$, we obtain $\hat{v}_i'(d_i) = \frac{1}{p_i} v_i'(d_i)(1 - d_i/C)$. Since v_i is concave and strictly increasing, we know that $v_i'(d_i) > 0$, and that v_i' is non-increasing. Because $1 - d_i/C$ is decreasing over $0 \leq d_i \leq C$, we conclude that $\hat{v}_i'$ is is nonnegative and strictly decreasing in d_i over $0 \leq d_i \leq C$, as required.

Step 2: *There exists a unique $\mathbf{d}$ and scalar ρ such that:*

$$\hat{v}_i'(d_i) = \frac{1}{p_i} v_i'(d_i)(1 - \frac{d_i}{C}) = \rho, \quad \textit{if } d_i > 0; \tag{25}$$

$$\hat{v}_i'(0) = \frac{1}{p_i} v_i'(0) \leq \rho, \quad \textit{if } d_i = 0; \tag{26}$$

$$\sum_{j=1}^{N} d_j = C. \tag{27}$$

*The vector $\mathbf{d}$ is the unique optimal solution to the optimization problem (**??**)-(**??**).* By Step 1, we know that the optimization problem (**??**)-(**??**) has a unique optimal solution. This optimal solution $\mathbf{d}$ is uniquely identified by the optimality conditions (25)-(26) and the constraint $\sum_{i=1}^{N} d_i \leq C$. Because each $\hat{v}_i$ is strictly increasing, the constraint must be tight and satisfies (27). Finally, because at least one d_i is strictly positive, ρ is uniquely determined by Equation (25).

Step 3: *If $(\mathbf{d}, \rho)$ satisfies (25)-(27), then $\mathbf{t} = \rho\mathbf{d}$ is a Nash equilibrium.* First, we show that at least two components of $\mathbf{d}$ is strictly positive and $\rho > 0$. From Equation (27), we know at least one component of $\mathbf{d}$ is strictly positive. If only one component $d_i > 0$, we know that $d_i = C$, and from Equation (25), $\rho = 0$. However, since $v_i'(0) > 0$, condition (26) cannot hold. Thus at least two components of $\mathbf{d}$ are strictly positive and $\rho > 0$ follows from Equation (25).

By Lemma 1, we only need to check the conditions (23)-(24). Using Equation (27), we rewrite $\mathbf{t} = \rho\mathbf{d}$ as:

$$\rho = \sum_{i=1}^{N} t_i/C; \qquad d_i = t_i C / \sum_{i=1}^{N} t_i.$$

By substituting the above into (25)-(26), we obtain the conditions (23)-(24), and therefore, $\mathbf{t}$ is a Nash equilibrium.

Step 4: *If $\mathbf{t}$ is a Nash equilibrium, then the corresponding resource allocation $\mathbf{d}$ and $\rho = \sum_{i=1}^{N} t_i/C$ are the unique solution to (25)-(27).* We can reverse the argument of Step 3. The uniqueness of $(\mathbf{d}, \rho)$ follows by Step 2.

Step 5: *There exists a unique Nash equilibrium $\mathbf{t}$, and the resource allocation $\mathbf{d}$ defined by (4) is the unique optimal solution to (**??**)-(**??**).* Existence follows by Steps 2 and 3, and uniqueness follows by Step 4 (since the transformation from $\mathbf{t}$ to $(\mathbf{d}, \rho)$ is one-to-one). Finally, that $\mathbf{d}$ is an optimal solution to (**??**)-(**??**) follows by Steps 2 and 4.

∎

Proof of Theorem 4: Let $\mathbf{t} = \frac{1}{k}\mathbf{t}^{\mathbf{p}}$. We want to prove that $\mathbf{t}$ is a Nash equilibrium of $\mathscr{M}^{\mathbf{q}}$. Since $\mathbf{t}^{\mathbf{p}}$ is a Nash equilibrium, by Lemma 1, at least two components of $\mathbf{t}^{\mathbf{p}}$ are strictly positive; and therefore, so as the vector $\mathbf{t}$.

By Lemma 1, we know that the Nash equilibrium $\mathbf{t}^{\mathbf{p}}$ satisfies the following conditions:

$$\frac{1}{p_i} v_i'\Big(\frac{t_i^{\mathbf{p}} C}{\sum_{j=1}^{N} t_j^{\mathbf{p}}}\Big)\Big(1 - \frac{t_i^{\mathbf{p}}}{\sum_{j=1}^{N} t_j^{\mathbf{p}}}\Big) = \frac{\sum_{j=1}^{N} t_j^{\mathbf{p}}}{C}, \text{ if } t_i^{\mathbf{p}} > 0;$$

$$\frac{1}{p_i} v_i'(0) \leq \frac{\sum_{j=1}^{N} t_j^{\mathbf{p}}}{C}, \qquad \text{if } t_i^{\mathbf{p}} = 0.$$

By substituting $p_i = \frac{1}{k} q_i$ and $t_i^{\mathbf{p}} = k t_i$ into the above, we obtain the following conditions:

$$\frac{1}{\frac{1}{k} q_i} v_i'\Big(\frac{k t_i C}{\sum_{j=1}^{N} k t_j}\Big)\Big(1 - \frac{k t_i}{\sum_{j=1}^{N} k t_j}\Big) = \frac{\sum_{j=1}^{N} k t_j}{C}, \text{ if } k t_i > 0;$$

$$\frac{1}{\frac{1}{k} q_i} v_i'(0) \leq \frac{\sum_{j=1}^{N} k t_j}{C}, \qquad \text{if } k t_i = 0.$$

After dividing k on both sides, we obtain the following:

$$\frac{1}{q_i} v_i'\Big(\frac{t_i C}{\sum_{j=1}^N t_j}\Big)\Big(1 - \frac{t_i}{\sum_{j=1}^N t_j}\Big) = \frac{\sum_{j=1}^N t_j}{C}, \text{ if } t_i > 0;$$

$$\frac{1}{q_i} v_i'(0) \leq \frac{\sum_{j=1}^N t_j}{C}, \qquad \text{if } t_i = 0.$$

Since $\mathbf{t}$ has at least two strictly positive components and satisfies the above stationarity conditions, by Lemma 1, we conclude that $\mathbf{t} = \frac{1}{k}\mathbf{t}^{\mathbf{p}}$ is a Nash equilibrium of $\mathscr{M}^{\mathbf{p}}$. By Theorem 1, $\mathbf{t}$ is also the unique Nash equilibrium $\mathbf{t}^{\mathbf{q}}$ of $\mathscr{M}^{\mathbf{p}}$.

Since $\mathbf{t}^{\mathbf{q}} = \frac{1}{k}\mathbf{t}^{\mathbf{p}}$, the proportional share rule of (4) gives the same resource allocation, i.e. $d(\mathbf{t}^{\mathbf{q}}) = d(\mathbf{t}^{\mathbf{p}})$. Each user i's payment under $\mathscr{M}^{\mathbf{q}}$ is $q_i t_i^{\mathbf{q}} = k p_i \frac{1}{k} t_i^{\mathbf{p}} = p_i t_i^{\mathbf{p}}$, which is the same as the payment under $\mathscr{M}^{\mathbf{p}}$. Since users have the same valuation and payment under both $\mathscr{M}^{\mathbf{p}}$ and $\mathscr{M}^{\mathbf{q}}$, they achieve the same amount utility as well. Thus, we conclude that $\mathscr{M}^{\mathbf{q}}$ is equivalent to $\mathscr{M}^{\mathbf{p}}$.

∎

Proof of Theorem 1: Let $\mathbf{t} = \mathbf{t}^{\mathbf{p}}$. We want to prove that $\mathbf{t}$ is a Nash equilibrium of $\mathscr{M}^{\mathbf{q}}$. Since $\mathbf{t}^{\mathbf{p}}$ is a Nash equilibrium, by Lemma 1, at least two components of $\mathbf{t}^{\mathbf{p}}$ are strictly positive; and therefore, so as the vector $\mathbf{t}$.

To prove $\mathbf{t}$ to be a Nash equilibrium of $\mathscr{M}^{\mathbf{q}}$, by Lemma 1, we need to show that the following conditions are satisfied:

$$\frac{1}{q_i} v_i'\Big(\frac{t_i C}{\sum_{j=1}^N t_j}\Big)\Big(1 - \frac{t_i}{\sum_{j=1}^N t_j}\Big) = \frac{\sum_{j=1}^N t_j}{C}, \text{ if } t_i > 0;$$

$$\frac{1}{q_i} v_i'(0) \leq \frac{\sum_{j=1}^N t_j}{C}, \qquad \text{if } t_i = 0.$$

Since $q_i = p_i$ for all $t_i^{\mathbf{p}} > 0$, the first equation is the same as the stationarity condition of (23) for $\mathbf{t}^{\mathbf{p}}$ being a Nash equilibrium. Since $q_i \geq p_i$ for all $t_i^{\mathbf{p}} = 0$, we want to show

$$\frac{1}{q_i} v_i'(0) \leq \frac{1}{p_i} v_i'(0) \leq \frac{\sum_{j=1}^N t_j}{C}, \qquad \text{if } t_i = 0.$$

The above condition is the same as the stationarity condition of (24) for $\mathbf{t}^{\mathbf{p}}$ being a Nash equilibrium. By Theorem 1, $\mathbf{t}$ is also the unique Nash equilibrium $\mathbf{t}^{\mathbf{q}}$ of $\mathscr{M}^{\mathbf{p}}$.

Since $\mathbf{t}^{\mathbf{q}} = \mathbf{t}^{\mathbf{p}}$, both mechanisms achieve the same resource allocation, i.e. $d(\mathbf{t}^{\mathbf{q}}) = d(\mathbf{t}^{\mathbf{p}})$. Each user i's payment under $\mathscr{M}^{\mathbf{q}}$ is

$$q_i t_i^{\mathbf{q}} = q_i t_i^{\mathbf{p}} = \begin{cases} p_i t_i^{\mathbf{p}}, & \text{if } t_i^{\mathbf{p}} > 0; \\ 0, & \text{if } t_i^{\mathbf{p}} = 0. \end{cases}$$

This is the same as the payment under $\mathscr{M}^{\mathbf{p}}$. Since users have the same valuation and payment under both $\mathscr{M}^{\mathbf{p}}$ and $\mathscr{M}^{\mathbf{q}}$, they achieve the same amount utility as well. Thus, we conclude that $\mathscr{M}^{\mathbf{q}}$ is equivalent to $\mathscr{M}^{\mathbf{p}}$.

∎

Proof of Theorem 2: Let $\mathbf{t} = \mathbf{d}$ and $\mathbf{p}$ be defined as in Equation (9). Since $\sum_{i=1}^N t_i = \sum_{i=1}^N d_i = C$, the resource allocation to the strategy profile $\mathbf{t}$ becomes $\mathbf{d}(\mathbf{t}) = \mathbf{d}$ under the proportional share rule (4). Now, we want to prove that $\mathbf{t}$ is a Nash equilibrium of $\mathscr{M}^{\mathbf{p}}$. Since $\mathbf{t} = \mathbf{d} \in \mathcal{D}$, at least two components of $\mathbf{t}$ are strictly positive.

To prove $\mathbf{t}$ to be a Nash equilibrium of $\mathscr{M}^{\mathbf{q}}$, by Lemma 1, we need to show that the following conditions are satisfied:

$$\frac{1}{p_i} v_i'\Big(\frac{t_i C}{\sum_{j=1}^N t_j}\Big)\Big(1 - \frac{t_i}{\sum_{j=1}^N t_j}\Big) = \frac{\sum_{j=1}^N t_j}{C}, \text{ if } t_i > 0;$$

$$\frac{1}{p_i} v_i'(0) \leq \frac{\sum_{j=1}^N t_j}{C}, \qquad \text{if } t_i = 0.$$

Since $\sum_{i=1}^N t_i = C$, the right hand sides of the above equal 1. By substituting $t_i = d_i$ and $p_i = v_i'(d_i)(1 - d_i/C)$ into the above, the left hand sides equal 1 too. By Theorem 1, $\mathbf{t} = \mathbf{d}$ is also the unique Nash equilibrium $\mathbf{t}^{\mathbf{p}}$ of $\mathscr{M}^{\mathbf{p}}$.

∎

Proof of Theorem 5: We provide a constructive proof by building a one-to-one mapping from $\mathcal{D}$ to a set $\tilde{\mathcal{P}} \subseteq \hat{\mathcal{P}}$.

Step 1: *We first construct* $\tilde{\mathcal{P}}$. We define a mapping $h : \mathcal{D} \to \mathbb{R}^N$ as the following:

$$h_i(d_i) = v_i'(d_i)(1 - \frac{d_i}{C}), \quad i = 1, \cdots, N.$$

We define a mapping $g : \mathcal{D} \to \hat{\mathcal{P}}$ as the following:

$$g_i(\mathbf{d}) = d_i(\mathbf{h}(\mathbf{d})) = \frac{h_i(d_i)}{\sum_{j=1}^N h_j(d_j)} = \frac{v_i'(d_i)(1-\frac{d_i}{C})}{\sum_{j=1}^N v_j'(d_j)(1-\frac{d_j}{C})}.$$

Since $\mathbf{d} \in \mathcal{D}$, $d_i < C$ for all i, and therefore $h_i(d_i) > 0$ for all i. We can verify $g(\mathbf{d}) \in \hat{\mathcal{P}}$ by showing $\sum_{i=0}^N g_i(\mathbf{d}) = 1$ and $g_i(\mathbf{d}) > 0$ for all i. We define $\tilde{\mathcal{P}}$ as the imagine of $\mathcal{D}$ under g, i.e. $\tilde{\mathcal{P}} = g(\mathcal{D}) \subseteq \hat{\mathcal{P}}$.

Step 2: *We show that g is a continuous mapping.* Because v_i is continuously differentiable, $h_i = v_i'(d_i)(1-d_i/C)$ is continuous over the domain $d_i \in [0, C)$. The resource allocation function $d_i(\mathbf{h})$ defined in (4) is also continuous over the domain $[h_i(0), h_i(C))$. Since g is a composite function of $d_i(\cdot)$s and $h_i(\cdot)$s, we conclude that g a continuous mapping from $\mathcal{D}$ to $\hat{\mathcal{P}}$.

Step 3: *g is onto $\tilde{\mathcal{P}}$.* This follows by the definition of $\tilde{\mathcal{P}} = g(\mathcal{D})$.

Step 4: *g is a one-to-one mapping.* For any $\mathbf{d} \in \mathcal{D}$, by Theorem 2, we know $\mathbf{d}(\mathbf{t}^{\mathbf{h}(\mathbf{d})}) = \mathbf{d}$. Because $g(\mathbf{d}) = kh(\mathbf{d})$ with $k = 1/\sum_{j=1}^N h_j(d_j)$, by Theorem 4, we know that $\mathbf{d}(\mathbf{t}^{g(\mathbf{d})}) = \mathbf{d}(\mathbf{t}^{h(\mathbf{d})}) = \mathbf{d}$. By Theorem 1, the mechanism $\mathscr{M}^{\mathbf{g}(\mathbf{d})}$ has a unique Nash equilibrium, and therefore $\mathbf{d}$ is the unique resource allocation in $\mathcal{D}$. Thus, we conclude that g is a one-to-one mapping.

Step 5: *There exists a continuous, one-to-one and onto mapping $f : \tilde{\mathcal{P}} \to \mathcal{D}$ such that $f(\mathbf{p}) = \mathbf{d}(\mathbf{t}^{\mathbf{p}})$.* By Step 2,3 and 4, g is continuous, one-to-one and onto. Therefore, it has a continuous, one-to-one and onto inverse mapping g^{-1}. Let $f = g^{-1}$ be the inverse mapping of g. By Step 4, we know that $\mathbf{d}(\mathbf{t}^{g(\mathbf{d})}) = \mathbf{d}$. Let $\mathbf{p} = g(\mathbf{d}) \in \tilde{\mathcal{P}}$ and substitute $\mathbf{p}$ into $\mathbf{d}(\mathbf{t}^{g(\mathbf{d})}) = \mathbf{d}$, we get $\mathbf{d}(\mathbf{t}^{\mathbf{p}}) = \mathbf{d} = g^{-1}(\mathbf{p}) = f(\mathbf{p})$.

Step 6: *$\tilde{\mathcal{P}}$ is connected.* This is because the imagine of f, i.e. $\mathcal{D}$, is connected and f is a continuous mapping.

Step 7: *$\tilde{\mathcal{P}} = \hat{\mathcal{P}}$ when $n = 2$.* By Theorem 1, any $\mathbf{p} \in \hat{\mathcal{P}}$ has a unique mapping in $\mathcal{D}$. We only need to show that there does not exist $\mathbf{p_1}, \mathbf{p_2} \in \hat{\mathcal{P}}$ with $\mathbf{p_1} \neq \mathbf{p_2}$ such that $f(\mathbf{p_1}) = f(\mathbf{p_2})$. When $n = 2$, $\mathcal{D} = \{\mathbf{d} \mid d_1 + d_2 = C, 0 < d_1, d_2 < C\}$, where d_1 and d_2 cannot be zero. Therefore, t_1 and t_2 cannot be zero in equilibrium. By Lemma 1, the following condition of (23) must be satisfied:

$$\frac{1}{p_1} v_1'\Big(\frac{t_1 C}{t_1+t_2}\Big)\Big(\frac{t_2}{t_1+t_2}\Big) = \frac{t_1+t_2}{C}$$
$$\frac{1}{p_2} v_2'\Big(\frac{t_2 C}{t_1+t_2}\Big)\Big(\frac{t_1}{t_1+t_2}\Big) = \frac{t_1+t_2}{C}$$

From the above equations, we get

$$\frac{p_1}{p_2} = \frac{v_1'\Big(\frac{t_1 C}{t_1+t_2}\Big)}{v_2'\Big(\frac{t_2 C}{t_1+t_2}\Big)} \frac{t_2}{t_1} = \frac{v_1'(d_1)}{v_2'(d_2)} \frac{d_2}{d_1}.$$

The above equation implies that for any $\mathbf{d} = (d_1, d_2) \in \mathcal{D}$, the price ratio p_1/p_2 is fixed. Thus, there does not exist $\mathbf{p_1}, \mathbf{p_2} \in \hat{\mathcal{P}} = \{p_1 + p_2 = 1, p_1, p_2 > 0\}$ with $\mathbf{p_1} \neq \mathbf{p_2}$ such that $f(\mathbf{p_1}) = f(\mathbf{p_2})$.

∎

Lemma *2 (Resource allocation):* For any $\mathbf{p} \in \mathcal{P}$ with the unique Nash equilibrium $\mathbf{t}^{\mathbf{p}}$, let $\rho = \sum_{i=1}^N t_i^{\mathbf{p}}/C$ and $h_i(d_i) = v_i'(d_i)(1-d_i/C)$ for all i. If $\mathbf{d} = \mathbf{d}(\mathbf{t}^{\mathbf{p}})$ is the resource allocation under $\mathscr{M}^{\mathbf{p}}$, then $\mathbf{d}$ is the unique solution that satisfies $\sum_{j=1}^N d_j = C$ and the following condition:

$$\frac{1}{p_i} h_i(d_i) \begin{cases} = \rho, & \text{if } d_i > 0; \\ \leq \rho, & \text{if } d_i = 0. \end{cases} \tag{28}$$

Proof of Lemma 2: The condition of (28) is equivalent to (25) and (26) in the proof of Theorem 1. This result follow the same argument of Step 4 in the proof of Theorem 1.

∎

Proof of Theorem 3: We first show that $h_i(d_i) = v_i'(d_i)(1-d_i/C)$ is a strictly decreasing function over the domain $d_i \in [0, C)$ for all i. v_i is a concave function, and therefore, v_i' is non-increasing. Since $(1-d_i/C)$ is a strictly decreasing function, $h_i(d_i)$ is a strictly decreasing function too. Thus, we have

$$h_i(d') > h_i(d) \Leftrightarrow d' < d, \quad \forall d, d' \in [0, C) \tag{29}$$

By Lemma 2, the following conditions must be satisfied:

$$\begin{cases} h_i(d_i) = p_i\rho, & \text{if } d_i > 0 \text{ and } i \neq k; \\ h_i(d_i) \leq p_i\rho, & \text{if } d_i = 0 \text{ and } i \neq k; \\ h_k(d_k) = p_k\rho, & \text{if } d_k > 0; \\ h_k(d_k) \leq p_k\rho, & \text{if } d_k = 0. \end{cases} \tag{30}$$

$$\begin{cases} h_i(d_i') = p_i\rho', & \text{if } d_i' > 0 \text{ and } i \neq k; \\ h_i(d_i') \leq p_i\rho', & \text{if } d_i' = 0 \text{ and } i \neq k; \\ h_k(d_k') = p_k'\rho', & \text{if } d_k' > 0; \\ h_k(d_k') \leq p_k'\rho', & \text{if } d_k' = 0. \end{cases} \tag{31}$$

$$\sum_{i=1}^{N} d_i = \sum_{i=1}^{N} d_i' = C. \tag{32}$$

Next, we prove the three mutually exclusive cases.

Case 1: $d_k > 0, p_k' < p_k$ or $d_k = 0, p_k' < h_k(0)/\rho$. From (30), we know that $p_k = h_k(d_k)/\rho$ for $d_k > 0$, and therefore, the first case is equivalent to $p_k' < h_k(d_k)/\rho$.

Step 1a: *We want to show that* $\rho' > \rho$. We use proof by contradiction. Suppose $\rho' \leq \rho$. For any $d_i > 0$, $i \neq k$, by (30) and (31), we have

$$h_i(d_i') \leq p_i\rho' \leq p_i\rho = h_i(d_i).$$

Since $h_i(d_i') \leq h_i(d_i)$, by (29), we know $d_i' \geq d_i$ for all $d_i > 0, i \neq k$. Since $d_i' \geq 0$ for all i, $d_i' \geq d_i$ for all $i \neq k$. For user k, by (30), (31) and the condition $p_k' < h_k(d_k)/\rho$, we have

$$h_k(d_k') \leq p_k'\rho' \leq p_k'\rho < (h_k(d_k)/\rho)\rho = h_k(d_k).$$

Since $h_k(d_k') < h_k(d_k)$, by (29), we know $d_k' > d_k$. However, by (32), we know that with $d_k' > d_k$, it is impossible to have $d_i' \geq d_i$ for all $i \neq k$. Thus, we conclude that $\rho' > \rho$.

Step 1b: *We show that* $d_i' < d_i$ *for* $d_i > 0, i \neq k$. If $d_i' = 0$, then $d_i' = 0 < d_i$. If $d_i' > 0$, by (30), (31) and $\rho' > \rho$, we have

$$h_i(d_i') = p_i\rho' > p_i\rho = h_i(d_i).$$

Since $h_i(d_i') > h_i(d_i)$, by (29), we know that $d_i' < d_i$.

Step 1c: *We show that* $d_i' = d_i$ *for* $d_i = 0, i \neq k$. Suppose $d_i' > d_i = 0$, by (30) and (31), we have

$$h_i(d_i') = p_i\rho' > p_i\rho \geq h_i(d_i) = h_i(0).$$

Since $h_i(d_i') > h_i(0)$, by (29), we would have $d_i' < 0$, which is not possible. Thus, we conclude that $d_i' = d_i = 0$ for $d_i = 0, i \neq k$.

Step 1d: *We show that* $d_k' > d_k$. By Lemma 1, we know at least two components of $\mathbf{t}^{\mathbf{P}}$ are strictly positive. By the proportional share rule of Equation (4), we know that at least two components of the resource allocation $\mathbf{d}$ are strictly positive. Therefore, at least one user $i \neq k$ has a strictly positive resource allocation d_i, and by Step 1b, $d_i' < d_i$. By Step 1b and 1c, we know that $d_i' \leq d_i$ for all $i \neq k$. By condition (32), we conclude that $d_k' > d_k$.

Case 2: $d_k > 0$ and $p_k' > p_k$.

Step 2a: *We want to show that* $\rho' < \rho$. Similar to Step 1a, we use proof by contradiction. Suppose $\rho' \geq \rho$. For any $d_i' > 0$, $i \neq k$, by (30) and (31), we have

$$h_i(d_i') = p_i\rho' \geq p_i\rho \geq h_i(d_i).$$

Since $h_i(d_i') \geq h_i(d_i)$, by (29), we know $d_i' \leq d_i$ for all $d_i' > 0, i \neq k$. Since $d_i' \leq 0$ for all i, $d_i' \leq d_i$ for all $i \neq k$. For user k with $d_k' > 0$, by (30), (31) and the condition $p_k' > p_k$, we have

$$h_k(d_k') = p_k'\rho' \geq p_k'\rho > p_k\rho = h_k(d_k).$$

Since $h_k(d_k') > h_k(d_k)$, by (29), we know $d_k' < d_k$. If $d_k' = 0$, we also have $d_k' < d_k$. However, by (32), we know that with $d_k' < d_k$, it is impossible to have $d_i' \leq d_i$ for all $i \neq k$. Thus, we conclude that $\rho' < \rho$.

Step 2b: *We show that* $d_i' > d_i$ *for* $p_i\rho' < h_i(d_i), i \neq k$. By (30), (31) and $p_i\rho' < h_i(d_i)$, we have

$$h_i(d_i') \leq p_i\rho' < h_i(d_i) \leq p_i\rho.$$

Since $h_i(d_i') < h_i(d_i)$, by (29), we know that $d_i' > d_i$.

Step 2c: *We show that* $d_i' = d_i = 0$ *for* $p_i\rho' \geq h_i(d_i), i \neq k$. First, we show that $d_i = 0$. Since $\rho' < \rho$, we have

$$p_i\rho > p_i\rho' \geq h_i(d_i) \Rightarrow h_i(d_i) < p_i\rho.$$

By (30), we know that $d_i = 0$. Suppose $d_i' > d_i = 0$, by (31), we have

$$h_i(0) = h_i(d_i) \leq p_i\rho' = h_i(d_i').$$

Since $h_i(d_i') \geq h_i(d_i)$, by (29), we would have $d_i' \leq d_i = 0$, which contradicts to $d_i' > d_i = 0$. Thus, we conclude that $d_i' = d_i = 0$ for $p_i\rho' \geq h_i(d_i), i \neq k$.

Step 2d: *We show that* $d'_k < d_k$. Similar to Step 1d, by Lemma 1, we know at least two components of $\mathbf{t^p}$ are strictly positive. By the proportional share rule of Equation (4), we know that at least two components of the resource allocation $\mathbf{d}$ are strictly positive. Therefore, at least one user $i \neq k$ has a strictly positive resource allocation d_i, and by Step 2b, $d'_i > d_i$. By Step 2b and 2c, we know that $d'_i \geq d_i$ for all $i \neq k$. By condition (32), we conclude that $d'_k < d_k$.

Case 3: $d_k = 0$ and $p'_k \geq h_k(0)/\rho$.

Step 3a: *Let* $d'_i = d_i$ *for all* i *and* $\rho' = \rho$*, we verify that condition (31) is satisfied.* This is because the conditions in (30) and (31) are the same for users $i \neq k$, and for user k, since $h_k(d_k) = h_k(d'_k)$ and $p'_k\rho' = p'_k\rho$, we have $h_k(d'_k) \leq p'_k\rho'$ because

$$h_k(d'_k) = h_k(d_k) \leq p'_k\rho = p'_k\rho'.$$

Step 3b: By Lemma 2, $d'_i = d_i$ is the unique resource allocation under $\mathscr{M}^{\mathbf{p}'}$

■

Proof of Theorem 6: This is simple result from the Nash equilibrium condition (23) for $t_i^{\mathbf{p}} > 0$. By substitute the resource allocation from Equation (4) into the Nash equilibrium condition (23), we get the above Equation (13). ■

Proof of Theorem 7: We start with the "only if" part. Because $\mathbf{d} > 0$, the optimality condition for $\mathbf{d}$ being optimal is

$$v'_i(d^*_i) = v'_j(d^*_j), \quad \forall i, j = 1, \cdots, N.$$

Suppose $\mathbf{p}$ induces a Nash equilibrium that achieves $\mathbf{d}$. Then the Nash equilibrium condition of (23) can be rewritten as:

$$\frac{1}{p_i} v'_i(d^*_i)(C - d^*_i) = \sum_{j=1}^{N} t_j^{\mathbf{p}}, \quad \forall i = 1, \cdots, N.$$

Therefore $p_i : p_j = v'_i(d^*_i)(C - d^*_i) : v'_j(d^*_j)(C - d^*_j) = C - d^*_i : C - d^*_j$ for all $i, j = 1, \cdots, N$.

Then we solve the "if" part. By Theorem 2, we know that $\mathbf{p}$ defined by Equation (9), i.e. $p_i = h_i(d^*_i) = v'_i(d^*_i)(1 - \frac{d^*_i}{C})$ for all $i = 1, \cdots, N$ induces d^*. Since $v'_i(d^*_i)$ is the same for any user i, we know $\mathbf{p}$ satisfies condition (14). Then any price vector $\hat{\mathbf{p}}$ that satisfies condition (14) can be expressed as $\hat{\mathbf{p}} = k\mathbf{p}$ for some positive constant k. By Theorem 4, we know $\hat{\mathbf{p}}$ achieves $\mathbf{d}^*$ as well. ■