INCENTIVE PROVISION AND JOB ALLOCATION IN SOCIAL CLOUD SYSTEMS

Yu Zhang, Mihaela van der Schaar

Abstract—Social cloud systems, which aggregate the computing capabilities of a large pool of users, have emerged in recent years as a key solution for resource provision and sharing in large-scale online communities due to their inherent flexibility and cost-effectiveness. However, the performance and reliability of these systems depend on the users' cooperative behavior in sharing their computing capabilities. Hence, incentive mechanisms are needed to deter users from free-riding. In this paper, we first model the selfish behavior of the users supplying resources and aiming to maximize their own benefits, and compute the performance of the resulting non-cooperative equilibrium. Since the non-cooperative equilibrium emerging when suppliers pursue their self-interests are highly inefficient, we augment the existing job allocation schemes currently implemented in social cloud systems with a novel class of incentive mechanisms based on reputation-based pricing and collective punishment schemes which compels suppliers to change their selfish strategies in a manner that improves the efficiency of the system. We study the cloud system operator's problem of jointly optimizing the incentive mechanism and the job allocation scheme in order to find an optimal social cloud protocol which eliminates the free-riding behavior of suppliers while maximizing the social welfare of the system. We rigorously prove that, using only simple designs for both the incentive mechanism and the job allocation scheme, the resulting protocol provides significant improvements in terms of the social welfare compared to existing social cloud systems.

I. INTRODUCTION

Social cloud systems are booming nowadays [1]. In a nutshell, a social cloud system can be viewed as an online trading market for various resources, where individual users join to contribute the underused resources of their devices (e.g. smartphones, personal computers, tablets) in exchange for monetary rewards [2]-[4]. Examples of exchanged resources include storage spaces, bandwidth, computing capacity (e.g. CPU cycles), multi-modal data, and even knowledge and information. Many social cloud systems are organized in the following manner. A user, who is referred to as a buyer, submits a task (e.g. the online storage of 20GB data for one day) together with the requested payment to a cloud operator, who is usually the owner of the market. The cloud operator then selects capable suppliers, who are also users in the system, and divides the task into small jobs and allocates the jobs among them. Upon receiving the payments, the selected suppliers start to jointly serve (i.e. provide their resources to) the buyer.

Traditional cloud computing systems commonly assume a server-client architecture with resources (e.g. data centers) being owned and centrally operated by an autonomous administrative body (e.g. Amazon, Google or Microsoft). Nevertheless, since the workloads are highly dynamic and heterogeneous, performing effective resource allocation and power management for large-scale enterprise data centers becomes very challenging, in order to minimize the maintenance cost of idle resources while ensuring elastic on-demand services of high-quality [5]-[10].

Alternatively, social cloud systems are constructed with peer-to-peer architectures with resources being owned and managed distributedly by individual users. Each user in the system can be both a supplier providing resources and a buyer requesting resources from other users. By aggregating the computing capabilities of a large pool of users, social cloud systems are able to provide a more flexible and cost-effective solution for resource provision compared to the traditional cloud computing systems.

A significant body of works was recently dedicated to constructing and operating social cloud systems. For instance, [2] presented a cloud-based approach for individual devices to exchange their underused resources to generate revenue. An alternative framework for social cloud computing was proposed in [3], where the friends within a social network can share their resources by dynamically forming a "social cloud". In [11], a peer-to-peer cloud for resource sharing was constructed and a reliable resource provision mechanism was proposed in order to enable churn resistant services.

The design of many social cloud systems, however, rely on the assumption that users are cooperative and altruistic in resource provision [2][3][11][12]. The fact that users in social cloud system are self-interested and aim to maximize their individual utilities is usually neglected. Therefore, these systems are affected by the "free-riding" behaviors of their users [16] which significantly degrade the overall performance. Given the self-interested nature of users, it is thus crucial to design appropriate incentives that encourage the user contribution to ensure the high performance of social cloud systems.

There are only few works studying the design of incentive mechanisms for social cloud systems. Most of these works rely on pricing schemes which incentivize suppliers to provide their resources by using monetary rewards [13]-[15]. Nevertheless, pricing schemes are not always effective in encouraging the contribution of resources from suppliers since they fail to address the following social dilemma existing between buyers and suppliers: if the payment associated with a task is performed ex-ante (i.e. suppliers receive payments before the task starts), a supplier always has the incentive to take the payment without providing the promised resources, i.e. to free-ride [16]; whereas if the payment is ex-post/ex-interim, which means that suppliers cannot receive payments until tasks are completed, the buyer always has the incentive to refuse payment by lying about the outcome of the task, a behavior commonly known as "false-reporting" [16]. Moreover, these works treat the interactions among suppliers and buyers as a one-time exchange and do not exploit the fact that such interactions are often repeated. Last but not least, the proposed incentive mechanisms in these works rely on the assumption that the individual behavior of each supplier can be accurately identified and measured. This assumption does not hold in many social cloud applications where each task is collectively performed by multiple suppliers. For example, in the social cloud system that provides mobile streaming services [17], each buyer only observes the aggregate bandwidth it receives, without knowing who provide this streaming service and how this service is divided among different mobile suppliers.

In this paper, we address these problems by designing a novel class of incentive mechanisms which rely on reputation schemes. We assume that the payment of each task is performed ex-ante and focus mainly on suppressing the "free-riding" behavior of suppliers by encouraging their resource contribution [1]. The underlying idea of our incentive mechanism can be described as follows. We assign to each supplier in a social cloud system a reputation. A supplier earns its reputation based on its past resource provision

[1] However, our protocol and system design can be straightforwardly extended to address the "false-reporting" of buyers.

behavior. A differential reputation-based reward/punishment scheme is performed by the cloud operator: it assigns higher payments to suppliers with higher reputations (i.e. suppliers who performed well in the past in their resource provision). Thus, suppliers are provided incentives to contribute their resources.

The differences between our work and the existing literature on cloud systems are listed in Table 1 in the appendix. In summary, the main novelty and contribution of our work lie in the following aspects:

(1) We explicitly formalize the interactions between suppliers and buyers using a rigorous repeated game framework and exploit such repeated nature in the design of incentive mechanisms.

(2) We rigorously develop social cloud protocols that integrate incentive mechanisms and job allocation schemes in a rigorous manner.

(3) We design a novel collective rating scheme to provide incentives, which does not require sophisticated technologies to monitor the suppliers' behaviors and is therefore for easy implementation in social cloud systems. This also extends the existing works on reputation system design for online communities [16][18][19], which rely on the assumption that the individual behavior of each supplier can be accurately identified and measured and are thus complex to be implemented in social cloud systems.

In the design of effective incentive mechanisms, we first capture the repeated interactions between buyers and suppliers within the formalism of continuous-time repeated games. To formalize the incentive mechanism, a novel social norm [19], which consists of a pricing scheme and a reputation scheme, is introduced to regulate the provision of resources by suppliers. Since the individual behavior of each supplier cannot be credibly identified, the reputation scheme updates the reputations of suppliers based on their collective resource contribution. By rigorously analyzing how the suppliers' equilibrium behavior will be influenced by the pricing and reputation schemes, we propose incentive mechanisms which can effectively eliminate the suppliers' free-riding behavior. We also systematically investigate the design of the job allocation scheme adopted by the cloud operator, which determines how a task should be divided into jobs and allocated among various suppliers. We then show how the cloud operator should define and solve the protocol design problem of jointly optimizing the incentive mechanism and the job allocation scheme in order to eliminate the free-riding behavior of suppliers and maximize the social welfare of the system. As an illustrative example, we discuss in detail the design of a class of homogeneous job allocation schemes. Such job allocation schemes are not only simple to design and easy to deploy, but also achieve a good performance that is close to the social optimum. Based on our knowledge, this paper is the first work that integrates incentive mechanisms and job allocation schemes in a rigorous manner to design protocols for social cloud systems.

The remainder of this paper is organized as follows. In Section II, a rigorous analytical framework is proposed to analyze the interactions emerging in social cloud systems. In Section III, we characterize the emerging non-cooperative equilibrium and formulate the optimal protocol design problem, which is explicitly solved in Section IV. After presenting the simulation results in Section V, we conclude in Section VI, where we also outline future research topics.

II. System Model

A. Setup

A social cloud system can be regarded as an online trading platform for various resources. In this paper, we focus solely on trading a single type of resources. Note though that the incentive mechanism and the job allocation scheme which are designed in this paper can be easily extended to the trading on multiple types of resources. The precise formulation of a social cloud system will depend on details of the type of resources being exchanged and the services being offered, the implementation infrastructure of the system, the characteristics of participating users, etc. In this paper, we use a stylized model to formulate the operations on a social cloud system without delving into the idiosyncrasies of any particular applications, in order to capture the basic trade-offs and draw qualitative insights about the effects of self-interested behavior on the system performance.

A social cloud system consists of users and cloud operators. Users can be further classified as *buyers* who initiate tasks to purchase resources from the system and *suppliers* who provide the purchased resources. Our definition of "supplier" is very general, and can encompass any individual processing nodes, mobile devices, data centers etc., whoever has unused resources that can be offered to the system in order to gain revenues. Suppliers are assumed to be long-lived in the system, i.e. their lifetimes in the system are much longer than the durations of tasks. It should be noted that a supplier can also be a buyer and purchase resources from the system, and vice versa. *Cloud operators*, simply referred to as *operators* in the rest of this paper, are the designers and operators of the system. They are responsible for designing and implementing (1) the pricing scheme and the payment infrastructure; (2) the incentive mechanism to regulate the behavior of suppliers; and (3) the job allocation scheme which schedules the load (the amount of resources purchased) of incoming tasks among suppliers. In this paper, we assume for simplicity that there is a single cloud operator.

We use a continuum model (mass 1) to describe the populations for both suppliers and buyers, which is a good practical model for large-scale social cloud systems [2]. The operator maintains and updates periodically the resource map that records the idle (unused) resources of each supplier which is available to be purchased by buyers. The tasks initiated by buyers arrive at the system following a Poisson process. When a buyer needs some resources, it submits a task to the operator. The load of a task, denoted as $K \in \mathbb{R}^+$, represents the total amount of resources needed to fulfill this task [3]. We assume that K takes

[2] In a continuum model, the impact of an individual user is negligible because the whole population is sufficiently large. Hence, the Law of Large Numbers can be applied.

[3] In some applications, e.g. online backup, the load (e.g. the storage space) is explicitly specified by the buyer when it submits the task, whereas in some other applications, e.g. scientific computing, the load (e.g. the cpu cycles needed to complete a computing task) is approximately estimated by the cloud operator. In general, the load of a task depends on its duration and the resources consumed per unit time. We assume that

values in $[K_{\min}, K_{\max}]$ with a distribution $g(K)$ [4] Hence, the system does not accept any task whose load is larger than $K_{\max}$ or smaller than $K_{\min}$. We also assume that both values $K_{\min}$ and $K_{\max}$ are predetermined and fixed. Hence, the system expects an average load of $\bar{K} = \int_{K_{min}}^{K_{max}} Kg(K)dK$ per task.

Upon receiving a task of load K, the operator selects a group of suppliers, who have sufficient idle resources to jointly fulfill the buyer's needs. This group of suppliers is then called as the working group for this task. The operator then determines the total price that the buyer has to pay for this task. After the submission of the payment, the operator divides the task, including its load and the associated payment, into multiple jobs with each supplier working on one job. Let G denote the working group with each supplier within it indexed with $i = \{1, ..., |G|\}$, the operator divides the load as $\{k_i\}_{i=1}^{|G|}$ where k_i is the load of the job assigned to the i-th supplier and $\sum_{i \in G} k_i = K$ [5]. Meanwhile, we also assume that the total payment allocated to the working group also equals the payment submitted by the buyer. Hence, the operator reserves no revenue for itself in a task. A detailed illustration of the operations in the system will be discussed in Section II.D when the incentive mechanism and the job allocation scheme are introduced.

B. Stage game

In this section, we characterize the interaction between a buyer and a supplier in one job. Such an interaction can be formulated as an asymmetric gift-giving game. Since the supplier receives its payment in advance, it can strategically choose its action when performing on the job, i.e. it strategically determines the amount of resources it provides to the buyer. If $a \in \mathbb{R}^+$ denotes the supplier's action, then the utility received by the supplier from one job can be written as $Q - ac$, where Q is the received payment (a detailed discussion about payments is provided in the next section) and c is the cost of providing unit resource. If there are no incentive mechanisms, a supplier always chooses $a = 0$ to free-ride, regardless of the amount of payments it receives, in order to myopically maximize its utility. Hence, a system will collapse with no resources being provided to buyers as a result of such free-riding behavior.

C. Incentive mechanisms

The reason why suppliers find it optimal to free-ride in the above stage game is the absence of punishments when they are not providing the promised resources. Note that, since suppliers and buyers are interacting repeatedly in the system by playing the above stage game with different opponents at each point

---

the durations of tasks are the same for better illustration. Such assumption encompasses the services provided in real systems. For example, Amazon EC2 sells its services on a monthly basis. It should be noted that whether durations of tasks are different does not affect the analysis.

[4] It should be noted here that K is not necessarily to be a continuous variable but could take only several discrete values within the region $[K_{\min}, K_{\max}]$. For example, an online storage service only accepts storage requests of 10GB, 20GB and 30GB. Our analysis still applies in this scenario with $g(K)$ consists of a series of the Dirac delta functions.

[5] In real systems, the operator should consider replications in job allocation in order to add certain redundancy and reduce the chance of task failure. However, this issue does not affect our analysis and hence we assume that tasks are allocated without replications in the analytical model.

of time, incentive mechanisms can be designed by the operator through the enforcement of punishments on suppliers' free-riding behaviors. To prevent free-ridings from suppliers, we deploy the idea of social norms [19] to construct the incentive mechanism. A social norms κ consists of a reputation label set Θ (used to record suppliers' past behavior), a reputation scheme τ (determining the rule of reputation update), and a pricing scheme ψ (realizing the reputation-based reward and punishment).

In the repeated game, each supplier is tagged with a reputation θ representing its social status, which is maintained and updated by the operator. We consider θ to be a natural number from the finite set $\Theta = \{0,1,2,\cdots,L\}$, where L is called the reputation length. A high reputation reflects a supplier's good behavior when performing on tasks it participates in the past.

The reputation scheme τ determines how the reputation of each supplier should be updated according to its past behavior. It should be noted that the interactions between buyers and suppliers (and thus, the actions of suppliers) cannot be directly monitored by the operator. Hence, the reputation scheme is operated solely depending on the buyers' submitted reports about whether their tasks are fulfilled (i.e. whether they are satisfied with the (joint) work of suppliers). Since a buyer is served jointly by a group of suppliers, its report on a task depends on the collective action performed by suppliers in the group. We assume that the format of a buyer's report is a binary variable $r \in \{0,1\}$, where a positive report $r = 1$ is submitted if and only if a buyer feels that its task is fulfilled; whereas a negative report $r = 0$ is submitted otherwise. Here we assume that each buyer has no incentive to misreport and always makes truthful reports.

Given truthful reporting, the report of a task is correlated to the collective action performed by suppliers in its working group through a conditional probability $p(r \mid \{a_i\}_{i=1}^{|G|}, K)$. If the load of a task is K, the buyer is unsatisfied and reports $r = 0$ whenever $\sum_{i \in G} a_i < K$, i.e. it receives less amount of resources than what it pays for. When $\sum_{i \in G} a_i = K$, the buyer might still be unsatisfied due to various reasons, e.g. caching and retrieving latencies, processing delays, networking errors etc. In this paper, we assume that such errors take place independently in each job with a small probability $\varepsilon \ll 1$. Therefore, the probability that a negative report is submitted after a task (i.e. the task fails) when $\sum_{i \in G} a_i = K$ is $\gamma = 1 - (1-\varepsilon)^{|G|}$. To summarize, we have: $p(r = 1 \mid \{a_i\}_{i=1}^{|G|}, K) = 1 - \gamma$ and $p(r = 0 \mid \{a_i\}_{i=1}^{|G|}, K) = \gamma$, if $\sum_{i \in G} a_i = K$; whereas $p(r = 1 \mid \{a_i\}_{i=1}^{|G|}, K) = 0$ and $p(r = 0 \mid \{a_i\}_{i=1}^{|G|}, K) = 1$ if $\sum_{i \in G} a_i < K$. Since γ depends on the size $|G|$ of the group of suppliers, its value might vary for different tasks. The detailed computation of γ will be discussed later when the job allocation scheme is introduced.

Given the report r from a task, τ updates the reputation for each supplier working on this task independently in the following manner: if $r=0$, each supplier's reputation falls to 0 with a probability β and remains unchanged with a probability $1-\beta$; whereas if $r=1$, each supplier's reputation is increased by 1 with a probability α and remains unchanged with a probability $1-\alpha$. Hence, the reputation scheme can be represented by a mapping as $\tau : \Theta \times \Theta \times \{0,1\} \to [0,1]$ and expressed as follows:

$$\text{if } r=1,\ \tau(\theta,\theta',r)=\begin{cases}\alpha & if\ \theta'{=}\min\{\theta{+}1,\mathrm{L}\}\\ 1-\alpha & if\ \theta'{=}\theta\end{cases};\ \text{if } r=0,\ \tau(\theta,\theta',r)=\begin{cases}\beta & if\ \theta'{=}0\\ 1-\beta & if\ \theta'{=}\theta\end{cases}. \tag{1}$$

Here θ represents a supplier's reputation at the beginning of the task and θ' represents its updated reputation after the task. A schematic representation of a reputation scheme is provided in Figure 1, where s is the output of a random device which generates numbers uniformly distributed over $[0,1]$.

Given the reputation scheme, the pricing scheme can be expressed as a mapping $\psi : \Theta \to \mathbb{R}^+$ and provides rewards/punishments to suppliers by assigning differentiated payments according to their reputations. With $\psi(\theta)$ increasing on θ, the incentive to free-ride is suppressed since a supplier is encouraged to provide resources in order to build up its reputation, which will lead to more payments in the future [6]. To keep the design and deployment of such a system simple, we restrict our attention to a simple class Ψ of cutoff pricing schemes. A cutoff pricing scheme $\psi \in \Psi$ can be characterized by a threshold $h(\psi) \in \{0,\ldots,L\}$ and is specified as: $\psi(\theta)=kq$ if $\theta \geq h(\psi)$, and $\psi(\theta)=0$ if $\theta < h(\psi)$, where q is the price for unit resource [7]. Briefly explained, under a cutoff pricing scheme, a supplier receives a payment q for each unit resource purchased by the buyer when its reputation is high. Here we assume q is exogenously determined by the operator and fixed in the analysis. Nevertheless, a supplier receives no payment in a job and has to work for free when its reputation is low. Therefore, the total payment that a buyer has to submit for a task of the load $K=\sum_{i\in G} k_i$ is $\sum_{\{i|\theta_i \geq h(\psi)\ and\ i\in G\}} qk_i$.

D. Job allocation scheme

Besides the incentive mechanism, the operator also has to design the job allocation scheme, i.e. to determine how to divide a task into multiple jobs and allocate them among suppliers. In this work, we assume that a task is always equally divided according to its load using a function $\varphi : \mathbb{R}^+ \to \mathbb{N}^+$. That is, a task of a load K is divided into $\varphi(K)$ job with each job having a load $K/\varphi(K)$. After the task division, the operator randomly selects $\varphi(K)$ suppliers, each of whom having at least an amount $K/\varphi(K)$ of idle

[6] It should be noted that such design of the reputation and pricing schemes is also robust to the possible churn [11] and strategic whitewashing [16] in the system. For example, the operator can always assign a low reputation to newly joined suppliers in order to prevent whitewashing.

[7] The analysis in this paper can be applied to cases when a general monotonically increasing pricing scheme is applied. The cutoff pricing scheme is applied here only for the illustration purpose.

resources, and assigns one job to each of them [8]. To simplify the analysis, we assume in the analytic model that the capacity of each supplier, i.e. the total amount of resources that each supplier can offer to the system is the same. However, it will be shown in Section V that this assumption is not necessarily needed as long as the capacity of each supplier is larger than certain value. The space of φ is denoted as Γ, and we consider the cardinality of Γ to be finite in our design.

To summarize, the design variables available to the operator is a tuple $\pi = (L, h, \alpha, \beta, \varphi)$, which we refer to as a social cloud protocol (SCP). The first four elements of an SCP represent the incentive mechanism, where L determines the *length of the history* recording a supplier's past behavior, h represents the *cutoff threshold* in the pricing scheme, α is the *forgiving probability*, and β is the *punishment probability*. The last element φ represents the job allocation scheme. In Table 2, we summarize the variables discussed in this paper for the better illustration. Given an SCP π, the operations incurred in a task are illustrated in Figure 2, where the numbers indicate the timely order of different operations.

In the rest of this paper, we analyze how to design the SCP in order to maximize the social welfare of the system, which is defined as the average revenue generated per task in the long run, under the condition that no supplier free-rides such that the system delivers a good quality of service. The detailed formulation of the SCP design problem will be presented in the next section.

III. DESIGNING SUSTAINABLE SCP

A. Suppliers' utilities and the social welfare

The strategy of a supplier in the repeated game can be formulated as a policy $\omega : \Theta \times \mathbb{R}^+ \to \mathbb{R}^+$. A strategic supplier determines its action in a job, i.e. the amount of resources it provides to the buyer, based on its strategy as $a = \omega(\theta, k)$, where θ is the supplier's reputation at the beginning of the job and k represents the amount of resources the buyer purchases from it. The utility of a supplier following a strategy ω is thus $v(\theta, k, \omega) = qk\psi(\theta) - c\omega(\theta, k)$. Regarding the fact that the arrival rate of tasks to the system follows a Poisson process and suppliers are randomly selected for each task, the job arrival to an individual supplier also follows a Poisson process with a rate λ_φ which depends on the job allocation scheme φ. Similarly, the load k of each job follows a distribution $f_\varphi(k)$ over a region $[k_{\min}^\varphi, k_{\max}^\varphi]$, which is jointly determined by $g(K)$ and φ with $k_{\min}^\varphi = \inf_{K \in [K_{min}, K_{max}]} K / \varphi(K)$ and $k_{\max}^\varphi = \sup_{K \in [K_{min}, K_{max}]} K / \varphi(K)$. A supplier's average utility from one task at reputation θ is thus $\overline{v}(\theta, \omega) = \int_{k_{\min}^\varphi}^{k_{\max}^\varphi} v(\theta, k, \omega) f_\varphi(k) dk$. Here we

[8] In the analytic model, the selection on suppliers is uniformly random, i.e. each supplier has a same probability to be selected as long as it has sufficient capacity to fulfill the allocated job. However, we also discuss more sophisticated selection rules (i.e. a supplier's chance to be selected depends on its reputation) in Section V. It turns out that the system does not have significant benefit with respect to the social welfare from such sophisticated selection rules. Hence, the random selection rule discussed here is sufficient in order to achieve close-to-optimal performance.

assume that after participating into a sufficient number of tasks, a supplier can correctly learn λ_φ and $f_\varphi(k)$ in order to calculate $\overline{v}(\theta,\omega)$. We use the infinite horizon discounted sum to evaluate a supplier's expected long-term utility by assuming that the supplier's appreciation on the future utility decays exponentially over time with a rate $\delta > 0$. Its *expected long-term utility* at the beginning of a task is

$$\begin{aligned} U(\theta^{(t_0)},k^{(t_0)},\omega) &= v(\theta^{(t_0)},k^{(t_0)},\omega) + \mathbb{E}(\sum_{i=1}^{\infty} e^{-\delta(t_i-t_0)} v(\theta^{(t_i)},k^{(t_i)},\omega)) \\ &= v(\theta^{(t_0)},k^{(t_0)},\omega) + (\int_{t_0}^{\infty} \lambda_\varphi e^{-\delta(t_1-t_0)} e^{-\lambda_\varphi(t_1-t_0)} dt_1) \sum_{\theta'} p_\omega(\theta' \mid \theta^{(t_0)},k^{(t_0)}) (\int_{k_{\min}^\varphi}^{k_{\max}^\varphi} U(\theta',k,\omega) f_\varphi(k) dk) \\ &= v(\theta^{(t_0)},k^{(t_0)},\omega) + \frac{\lambda_\varphi}{\lambda_\varphi+\delta} \sum_{\theta'} p_\omega(\theta' \mid \theta^{(t_0)},k^{(t_0)}) V(\theta',\omega) \end{aligned} \quad (2)$$

Here $\{t_i\}_{i=0}^{\infty}$ represents the timestamps when jobs occur at this supplier. $p_\omega(\theta' \mid \theta,k)$ denotes the expected transition probability of the reputation from θ to θ' after a job is finished when the supplier follows the strategy ω. $V(\theta,\omega) = \int_{k_{\min}^\varphi}^{k_{\max}^\varphi} U(\theta,k,\omega) f_\varphi(k) dk$ is the supplier's *average long-term utility* at reputation θ, which can be computed using the following recursive equation [9]:

$$V(\theta,\omega) = \overline{v}(\theta,\omega) + \frac{\lambda_\varphi}{\lambda_\varphi+\delta} \sum_{\theta'} p_\omega(\theta' \mid \theta) V(\theta',\omega). \quad (3)$$

Here $p_\omega(\theta' \mid \theta) = \int_{k_{\min}^\varphi}^{k_{\max}^\varphi} p_\omega(\theta' \mid \theta,k) f_\varphi(k) dk$. It should be noted that both $p_\omega(\theta' \mid \theta,k)$ and $p_\omega(\theta' \mid \theta)$ depend on the error probability γ in a task, which is again dependent on the size of the working group. By simply observing the load k for its own job, it is difficult for a supplier to correctly infer the load of the task and thus the group size. In this paper, we assume that each supplier knows $\overline{K}$ when it joins the system (e.g. it could be announced by the cloud operator) and each supplier maintains a belief that the expected group size for each task is approximately $n_k \triangleq \overline{K} / k$, when it receives a job with a load k [10].

Each supplier is foresighted, meaning that it selects its strategy such that its expected long-term utility at any moment is maximized. Since a supplier cannot observe the strategies of other suppliers, it maintains a simple belief that any other supplier is always cooperative in fulfilling its jobs. Given this belief, the optimal strategy of a supplier preserves the following structure:

---

[9] It should be noted that $U(\theta,k,\omega)$, $V(\theta,\omega)$, and $p_\omega(\theta' \mid \theta,k)$ also depend on π. Here we neglect such dependence in the notation for increased readability wherever it brings no confusion.

[10] We would like to point out that our analysis is not restricted to this particular belief model of suppliers but can generally apply to various belief models. For example, a supplier can believe that the size of the working group for each task stay the same. Our current framework can still be used to design effective SCPs to induce cooperation among suppliers under alternative belief models.

Definition 1 (Optimal Strategy). The optimal strategy ω_π^* given an SCP π is a strategy that preserves the property: $U(\theta,k,\omega_\pi^*) \geq U(\theta,k,\omega)$, $\forall\theta \in \Theta$ and $\forall k \in [k_{\min}^{\varphi}, k_{\max}^{\varphi}]$.

Remark: It should be noted that since suppliers maintain the same belief and are operating under the same SCP, they have the same optimal strategy. Hence in the derivation of the optimal strategy, we can focus on the decision problem of one typical supplier in the system. However at any moment when the system is operating, different suppliers might choose different actions based on the same optimal strategy, due to the difference on their reputations and the job loads. By analyzing (2) and (3), it is easy to determine that given an SCP π, a supplier's decision problem can be formulated as a continuous-time Markov decision process [20] where the state is (θ,k) and the action is a. Hence, the optimal strategy complies with the one-shot deviation principle [19]. That is, if a strategy ω_π^* is optimal, a supplier cannot benefit by deviating from ω_π^* in one job and complying with ω_π^* in all the subsequent jobs. Using this idea, the optimal strategy ω_π^* can be computed using dynamic programming and we have the following proposition that characterizes the structure of the optimal strategy.

Proposition 1. Given the SCP π, the optimal strategy ω_π^* for suppliers is *unique* and preserves the properties: (i) for each (θ,k), $\omega_\pi^*(\theta,k) \in \{0,k\}$; (ii) $\omega_\pi^*(\theta,k) \leq \omega_\pi^*(\theta+1,k)$, $\forall\theta \in \Theta$ and $\forall k \in [k_{\min}^{\varphi}, k_{\max}^{\varphi}]$.

Proof Sketch: For the proofs that are omitted or only provide sketches in this paper, the complete proofs and computations can always be found in the online appendix [22].

Since the decision process of a supplier can be formulated as a Markov decision process, the uniqueness of the optimal strategy can be straightforwardly proved using the convergence property of dynamic programming [21]. To prove the threshold-based property, it should first be noted that if a supplier free-rides and plays $a < k$, the future punishment a supplier expects to receive remains the same regardless of the selection on a. Hence, when a supplier decides to deviate, it is always optimal to provide no resource at all in order to maximize its gain from deviating in the current job. Then given the cut-off pricing scheme, it can be derived that a supplier's incentive to provide the requested resource monotonically increases with its reputation, due to the fact that the punishment it faces on free-riding increases with its reputation. ■

There are two important observations from Proposition 1. First, at any reputation θ, a supplier's action choice in one job is always binary, i.e. either cooperate by providing the full amount of resources purchased by the buyer or free-ride without providing any resource. Second, given a job of a load k, the current gain of deviation at any reputation is the same, whereas the future loss monotonically increases with the reputation θ. A supplier's incentive to free-ride thus monotonically decreases with its reputation.

The social welfare of the system is defined as the average revenue generated per task in the long-run. According to the pricing scheme in Section II.C, the social welfare is influenced by the reputation distribution of suppliers in the system, which is denoted by a vector $\boldsymbol{\eta} = \{\eta(\theta)\}_{\theta=0}^{L}$, where each term $\eta(\theta)$ represents the fraction of suppliers in the total population holding a reputation θ. Once a supplier's strategy converges to ω_π^*, its reputation evolves over time as a continuous-time Markov chain with the transition probability $p_{\omega_\pi^*}(\theta' \mid \theta)$ and hence, $\boldsymbol{\eta}$ evolves dynamically over time. In the long-run, $\boldsymbol{\eta}$ converges to a stationary point $\boldsymbol{\eta}_{\omega_\pi^*}$ which, according to the Law of the Large Numbers, is:

$$\eta_{\omega_\pi^*}(\theta) = \sum_{\theta' \in \Theta} p_{\omega_\pi^*}(\theta \mid \theta')\eta_{\omega_\pi^*}(\theta'),\ \forall \theta \in \Theta. \tag{4}$$

Therefore, the social welfare of the system can be expressed as follows, with its maximum value, defined as the social optimum, to be $q\bar{K}$.

$$W_\pi = \int_{K_{min}}^{K_{max}} \sum_{\theta \geq h} \eta_{\omega_\pi^*}(\theta) q \frac{K}{\varphi(K)} \varphi(K) g(K) dK = \sum_{\theta \geq h} \eta_{\omega_\pi^*}(\theta) q\bar{K}. \tag{5}$$

B. Sustainable SCP

According to Proposition 1, an arbitrarily selected SCP π might not effectively suppress the free-riding behavior of suppliers when the incentive it provides for cooperation is low. We first discuss how to design an SCP under which no supplier finds it in its self-interest to free-ride, i.e. *sustainable* SCP.

Definition 2 (Sustainable SCP). An SCP π is sustainable if and only if it induces ω_{coop} as the optimal strategy for suppliers, where $\omega_{coop}(\theta, k) = k$, $\forall \theta \in \Theta$ and $\forall k \in [k_{\min}^{\varphi}, k_{\max}^{\varphi}]$.

Under a sustainable SCP, providing the full amount of purchased resources in a job is always individually optimal for each supplier at any moment. In the rest of this section, we derive the sufficient and necessary conditions for an SCP π to be sustainable with $\omega_\pi^* = \omega_{coop}$.

Under a sustainable SCP, the reputation transition probability is determined as follows: $p_{coop}(\theta' \mid \theta, k)$ is $(1-\gamma_k)\alpha$ if $\theta' = \min\{\theta+1, L\}$, $\gamma_k \beta$ if $\theta' = 0$, and $(1-\gamma_k)(1-\alpha) + \gamma_k(1-\beta)$ if $\theta' = \theta$. Here, $\gamma_k = 1-(1-\varepsilon)^{n_k}$ represents the expected probability that a supplier will be punished upon cooperation. Since the supplier believes that any supplier other than itself plays with the strategy ω_{coop}, its expected long-term utility by complying with the strategy ω_{coop} in a job, given its reputation θ and the load k, is $U_{coop}(\theta, k) \triangleq qkI(\theta \geq h) - ck + \frac{\lambda_\varphi}{\delta + \lambda_\varphi} \sum_{\theta'} p_{coop}(\theta' \mid \theta, k) V(\theta, \omega_{coop})$, where $I(\bullet)$ is the indicator function . On the other hand, if the supplier decides to free-ride in this job by playing $a = 0$, its expected long-term utility becomes $U_{dev}(\theta,k) = qkI(\theta \geq h) + \frac{\lambda_\varphi}{\delta + \lambda_\varphi}((1-\beta)V(\theta,\omega_{coop}) + \beta V(0,\omega_{coop}))$. According to the one-shot deviation principle, an SCP is sustainable if and only if $U_{coop}(\theta,k) \geq U_{dev}(\theta,k)$. Since this argument should hold for any load k and any reputation θ, the sufficient and necessary condition for an SCP to be sustainable is determined as follows.

Proposition 2. An SCP π is sustainable if and only if

$$kc \leq \frac{\lambda_\varphi}{\delta + \lambda_\varphi}(1-\gamma_k)[\alpha V(\min\{\theta+1, L\}, \omega_{coop}) + (\beta - \alpha)V(\theta, \omega_{coop}) - \beta V(0, \omega_{coop})],$$

$$\text{and } \forall k \in [k_{\min}^\varphi, k_{\max}^\varphi]. \quad (6)$$ ■

C. Defining the optimal design problem

In this section, we discuss the design of a sustainable SCP that maximizes the social welfare of the system which is defined in (5). It should be noted that since $\bar{K}$ and q are predetermined and independent to the design of the SCP, the maximization on the social welfare is equivalent to the maximization on $\sum_{\theta \geq h} \eta_{\omega_{coop}}(\theta)$, i.e. the fraction of suppliers who have good reputations and can receive payments from their jobs. Therefore, the design problem can be re-expressed as follows:

$$\max_\pi \sum_{\theta \geq h} \eta_{\omega_\pi^*}(\theta), \quad s.t.\ \omega_\pi^* = \omega_{coop}. \quad (7)$$

Theorem 1 (Two-level Reputation is Optimal). Denote the solution of (7) to be π^* and the corresponding social welfare to be W^*, there is always an SCP with $L = h = 1$ that achieves W^*.

Proof Sketch: Let $\Pi_{\theta,k} = \frac{\lambda_\varphi}{\delta + \lambda_\varphi}\frac{1-\gamma_k}{k}[\alpha V(\min\{\theta+1, L\}, \omega_{coop}) + (\beta - \alpha)V(\theta, \omega_{coop}) - \beta V(0, \omega_{coop})]$ denote a supplier's incentive for cooperation in a job of load k and at reputation θ. It is obvious that the supplier chooses $a = k$ if and only if $\Pi_{\theta,k} \geq c$. Similar to Proposition 1, it can be proved that $\Pi_{\theta,k} \leq \Pi_{\theta+1,k},\ \forall \theta, k$. That is, the incentive for cooperation monotonically increases with the reputation. Hence, it is always true that an SCP is sustainable if and only if $\Pi_{0,k} \geq c,\ \forall k$. It can be proved that given L, α, β and φ, the pricing threshold $h = 1$ maximizes $\Pi_{0,k},\ \forall k$. An intuitive explanation for this is that with a smaller h, a supplier of reputation 0 could expect to restore its reputation and get paid sooner in the future and hence, its incentive to cooperate in order to restore its reputation becomes larger. Meanwhile, it is easy to prove that $\sum_{\theta \geq h} \eta_{\omega_\pi^*}(\theta)$ is maximized when $h = 1$ since only suppliers of reputation 0 are not paid in this case. Therefore, $h = 1$ is always optimal, when other design parameters are fixed. ■

Theorem 1 proves that the optimal social welfare can always be achieved with an incentive mechanism with two-level reputations [11]. Hence, it is sufficient for us to consider two-level reputation schemes and focus on the design of α, β and φ in the rest of this paper. For notational convenience, we use the concise expression $\pi = (\alpha, \beta, \varphi)$ in the analysis and denote the high reputation as $\theta = H$ and the low reputation as $\theta = B$. The design problem (7) thus can be rewritten as

$$\max_{\alpha,\beta,\varphi} \quad \eta_{\omega_\pi^*}(H), \qquad s.t.\ \omega_\pi^* = \omega_{coop}. \tag{8}$$

IV. OPTIMAL SCP DESIGN

In this section, we investigate the design of the optimal SCP, i.e. the optimal incentive mechanism and the optimal job allocation scheme, in order to solve the problem (8). For an improved presentation, the expressions of useful variables under a two-level reputation scheme are discussed below.

Given an SCP π and when $\omega_\pi^* = \omega_{coop}$, a supplier's expected reputation transition probability can be expressed as $p_{coop}(H \mid H, k) = 1 - \gamma_k + \gamma_k(1-\beta)$, $p_{coop}(B \mid B, k) = (1-\gamma_k)(1-\alpha) + \gamma_k$. Meanwhile, $\eta_{\omega_\pi^*}(H) = \dfrac{p_{coop}(H \mid B)}{p_{coop}(H \mid B) + p_{coop}(B \mid H)}$. Correspondingly, the average long-term utilities defined in (3) become $V(H, \omega_{coop}) = \bar{k}_\varphi (q - c) + \dfrac{\lambda_\varphi}{\lambda_\varphi + \delta}(p_{coop}(H \mid H) V(H, \omega_{coop}) + p_{coop}(B \mid H) V(B, \omega_{coop}))$ and $V(B, \omega_{coop}) = -\bar{k}_\varphi c + \dfrac{\lambda_\varphi}{\lambda_\varphi + \delta}(p_{coop}(H \mid B) V(H, \omega_{coop}) + p_{coop}(B \mid B) V(B, \omega_{coop}))$, where $\bar{k}_\varphi = \int_{k_{\min}^\varphi}^{k_{\max}^\varphi} f_\varphi(k) k dk$. We also have

$$V(H, \omega_{coop}) - V(B, \omega_{coop}) = \bar{k}_\varphi q \Big/ \left\{ 1 - \frac{\lambda_\varphi}{\lambda_\varphi + \delta}[(1 - \bar{\gamma}_\varphi)(1-\alpha) + \bar{\gamma}_\varphi (1-\beta)] \right\}, \tag{9}$$

where $\bar{\gamma}_\varphi = \int_{k_{\min}^\varphi}^{k_{\max}^\varphi} f_\varphi(k) \gamma_k dk$ is the average failure probability of a task given a job allocation scheme φ.

We discuss the optimal design in the following two steps. In the first step, we analyze given a job allocation scheme φ, how to optimally design the incentive mechanism, i.e. α and β. The corresponding solution are then denoted as α_φ and β_φ. In the second step, we design the optimal φ such that the SCP $\pi = (\alpha_\varphi, \beta_\varphi, \varphi)$ maximizes $\eta_{\omega_\pi^*}(H)$ with $\omega_\pi^* = \omega_{coop}$, which solves (8).

A. Optimal design of the incentive mechanism

---

[11] It should be noted that the optimal social welfare can also be achieved by some protocol with $L \geq h \geq 1$, and the protocol discussed in Theorem 1 is just one of the many optimal solutions.

We first discuss the existence of sustainable SCPs: given φ, whether there exist feasible values of α and β such that the SCP is sustainable. Based on Proposition 2 and (9), we have the following result.

Proposition 3. Given a job allocation scheme φ, sustainable SCPs exist if and only if

$$\frac{\lambda_\varphi}{\delta+\lambda_\varphi}\overline{k}_\varphi q \geq \max_{k\in[k_{\min}^\varphi,k_{\max}^\varphi]}\left(\frac{kc}{1-\gamma_k}\right). \qquad (10)\ \blacksquare$$

Hence, it can be determined from Proposition 3 that in order to design a sustainable SCP, a task should be divided into jobs whose loads are sufficiently small. However, a task also cannot be divided into too many jobs since γ_k increases as the division becomes finer. Given the existence of sustainable SCPs given a job allocation scheme φ, the optimal design α_φ and β_φ can be determined as follows.

Theorem 2 (Optimal Incentive Mechanism). Given a job allocation scheme φ that satisfies (10), the optimal design of the incentive mechanism is $\alpha_\varphi = 1$ and $\beta_\varphi = \max_{k\in[k_{\min}^\varphi,k_{\max}^\varphi]}\left\{\left(\frac{\delta+\lambda_\varphi}{\lambda_\varphi}\overline{k}_\varphi q - \gamma_k\overline{k}_\varphi q\right) / \left((1-\gamma_k)\overline{k}_\varphi q - \gamma_k kc\right)\right\}$.

Proof: According to (9), the optimal α and β can be solved with the linear programming problem:

$$\begin{aligned}
&\max_{\alpha,\beta}\frac{(1-\overline{\gamma}_\varphi)\alpha}{(1-\overline{\gamma}_\varphi)\alpha+\overline{\gamma}_\varphi\beta}\\
&s.t.\quad \frac{kc}{1-\gamma_k}\leq\frac{\lambda_\varphi}{\delta+\lambda_\varphi}\beta\frac{\overline{k}_\varphi q}{1-\frac{\lambda_\varphi}{\lambda_\varphi+\delta}[(1-\overline{\gamma}_\varphi)(1-\alpha)+\overline{\gamma}_\varphi(1-\beta)]},\forall k\\
&\qquad\frac{kc}{1-\gamma_k}\leq\frac{\lambda_\varphi}{\delta+\lambda_\varphi}\alpha\frac{\overline{k}_\varphi q}{1-\frac{\lambda_\varphi}{\lambda_\varphi+\delta}[(1-\overline{\gamma}_\varphi)(1-\alpha)+\overline{\gamma}_\varphi(1-\beta)]},\forall k\\
&\qquad 0\leq\alpha\leq 1,\ 0\leq\beta\leq 1
\end{aligned}, \qquad (11)$$

which leads to the solution in the statement of the theorem. ■

An interesting observation from Theorem 2 is that given a job allocation scheme φ, it is always optimal to assign suppliers with the highest forgiving probability, i.e. $\alpha = 1$, by upgrading their reputations to H whenever the report of the task is positive. In this way, suppliers of reputation B have the largest incentive for cooperation. On the other hand, β_φ should be the value which is just small enough to prevent a supplier from free-riding in a job.

B. Optimal design of the service allocation scheme

Given the optimal design α_φ and β_φ, the design problem (8) is now equivalent to the following optimization problem

$$\max_{\varphi\in\Gamma} \int_{k_{\min}^{\varphi}}^{k_{\max}^{\varphi}} f_{\varphi}(k)\frac{1-\gamma_k}{1-\gamma_k+\beta_{\varphi}\gamma_k}dk, \quad s.t.\ \frac{\lambda_{\varphi}}{\delta+\lambda_{\varphi}}\overline{k}_{\varphi}q \geq \max_{k\in[k_{\min}^{\varphi},k_{\max}^{\varphi}]}\left(\frac{kc}{1-\gamma_k}\right). \tag{12}$$

This problem is difficult to tract analytically. In Section V, we will solve it and illustrate the performance of the optimal SCP using integer programming methods. In the following proposition, we first provide some characterizations on the optimal job allocation scheme, denoted as φ^*.

Proposition 4. The optimal job allocation scheme φ^* preserves the following property:

$$\varphi^*(K) \leq \varphi^*(K'),\ \forall K, K' \in [K_{\min}, K_{\max}] \text{ and } K < K'. \tag{13}$$

Proof Sketch: According to (10), it can be shown that there are two regions $[\underline{k}_B, \overline{k}_B]$ and $[\underline{k}_H, \overline{k}_H]$ such that a supplier has sufficient incentive to cooperate if and only if the job load $k \in [\underline{k}_B, \overline{k}_B]$ when its reputation is B and $k \in [\underline{k}_H, \overline{k}_H]$ when its reputation is H. Hence, for a task of load K, we should have $K / \varphi^*(K) \leq \min\{\overline{k}_B, \overline{k}_H\}$ and $K / (\varphi^*(K)-1) > \min\{\overline{k}_B, \overline{k}_H\}$. Suppose $\varphi^*(K) > \varphi^*(K')$ with $K < K'$, it is obvious that $K' / \varphi^*(K') \geq K' / (\varphi^*(K)+1) > K / (\varphi^*(K)+1) > \min\{\overline{k}_B, \overline{k}_H\}$. Hence, suppliers will not cooperate in this task, which contradicts the claim that $\varphi^*(K')$ is the optimal job allocation for this task. ■

The reason why Proposition 4 holds is that in order to maximize the social welfare, the operator should assign to each individual supplier a job whose load is as large as possible (while still providing this supplier sufficient incentive to cooperate). Therefore, in the optimal job allocation scheme, the operator always assigns a larger working group for a task with a larger load, and φ^* then can be expressed as a monotonically increasing step function.

C. Optimal SCP design with homogeneous job allocation

As discussed in Section IV.B, the problem (12) of the optimal job allocation scheme is difficult to solve due to the large design space of φ. In this section, we investigate the homogeneous job allocation schemes, with which each task is always divided and assigned to a constant number of suppliers to work on, regardless of its load. An SCP with homogeneous job allocation is called SCPHA. We particularly show in this section that by focusing on such a subset of SCPs, the optimal protocol design problem becomes analytically tractable and easier to solve, while suffering little performance loss as the optimally designed SCPHA could deliver a good performance that is close to the optimal social welfare W^*.

Definition 3 (Homogeneous Job Allocation). A homogeneous job allocation scheme φ preserves the property that $\varphi(K) = n$, $\forall K \in [K_{\min}, K_{\max}]$.

For notational convenience, we use n to denote a homogeneous job allocation scheme and an SCPHA becomes $\pi = (\alpha, \beta, n)$. For the analysis, we also define several auxiliary variables: $\gamma(n) \triangleq 1-(1-\varepsilon)^n$ is the failure probability of each task, and $\lambda(n)$ is the arrival rate of tasks at each supplier. Obviously, $\lambda(n)$ is a linear function of n as $\lambda(n) = n\lambda(1)$. We also have the load of each job follows a distribution $f_n(k) = ng(nk)$ over the region $[k_{\min}^n, k_{\max}^n]$, with $k_{\min}^n = K_{\min} / n$ and $k_{\max}^n = K_{\max} / n$. Finally, we have $\bar{k}(n) \triangleq \bar{K} / n$ to be the average load of each job. Similar to Section IV.A, we first investigate the existence condition for sustainable SCPHAs.

Theorem 3 (Sustainable Region). Sustainable SCPHAs exist if and only if the system setting satisfies

$$\frac{c}{q} \le \frac{\bar{K}}{K_{\max}} \max_n \left\{ \frac{\lambda(n)}{\delta + \lambda(n)} (1 - \gamma(n)) \right\}. \tag{14}$$

Proof Sketch: First, it can be proved that by setting $\alpha = \beta = 1$, a supplier's incentive for cooperation is maximized. Substituting this into the constraints of (11) with $\bar{\gamma}_\varphi = \gamma(n)$, we can derive (14). ■

Intuitively explained, Theorem 3 shows that sustainable SCPHAs exist if and only if (i) the unit cost c is sufficiently small compared with the unit price q; (ii) the error probability ε for each job is sufficiently small; (iii) suppliers are sufficiently patient with a small discount factor δ; and (iv) the ratio $K_{\max} / \bar{K}$ is sufficiently small, i.e. the loads of incoming tasks do not change drastically over time. Theorem 3 provides important guidance for the operator in setting up the system, i.e. determine the price q and the values of $K_{\min}$ and $K_{\max}$. For instance, when the unit cost c and the error probability ε are large, the operator should select a larger price q in order to attract suppliers to contribute resources without free-riding. Also, smaller $K_{\max}$ should be selected in this case, which indicates that the system cannot accept tasks with too large loads in order to prevent free-riding. Next, we discuss the optimal design of sustainable SCPHAs. Similar to Theorem 2, we have the following result:

Theorem 4 (Optimal Incentive Mechanism with Homogeneous Job Allocation). When sustainable SCPHAs exist with a homogeneous job allocation scheme n, the optimal design α_n and β_n can be determined as $\alpha_n = 1$ and $\beta_n = \dfrac{\frac{\delta + n\lambda(1)}{n\lambda(1)} \bar{K}q - \gamma(n)\bar{K}q}{(1-\gamma(n))\bar{K}q - \gamma(n)K_{max}c}$. ■

With α_n and β_n, the optimal homogeneous job allocation scheme n can be solved as follows:

$$\max_n \quad \frac{1-\gamma(n)}{1-\gamma(n)+\gamma(n)\beta_n}, \quad s.t.\ 0 \le \beta_n \le 1. \tag{15}$$

In the next theorem, we obtain lower and upper bounds for the optimal social welfare that can be achieved using sustainable SCPHA. It should be noted that since sustainable SCPHAs constitute a subset of sustainable SCPs, these bounds also apply to the optimal social welfare W^*.

Theorem 5 (Bounds for Optimal Social Welfare). When sustainable SCPHAs exist, the optimal social welfare W^* is bounded as follows:

$$(1-\varepsilon)^{\ln(\frac{K_{max}c}{\bar{K}q+K_{max}c})/\ln(1-\varepsilon)}\bar{K}q \le W^* \le \bar{K}q\,. \quad (16)$$

Proof Sketch: When sustainable SCPHA exist, there is always an sustainable SCPHA with $\alpha = \beta = 1$, given the existence of sustainable SCPs. Given the job allocation to be n, the resulting social welfare of this SCPHA is $(1-\varepsilon)^n \bar{K}q$. It is proved in the online appendix that $n \le \ln(\frac{K_{max}c}{\bar{K}q + K_{max}c}) / \ln(1-\varepsilon)$ should hold in order to ensure that the resulting SCPHA is sustainable. Hence this social welfare is larger than or equal to $(1-\varepsilon)^{\ln(\frac{K_{max}c}{\bar{K}q+K_{max}c})/\ln(1-\varepsilon)}\bar{K}q$, which gives us the lower bound. The upper bound can be trivially proved and is omitted here. ■

In the next section, we explicitly compare the optimal sustainable SCP which is solved in (12) and the optimal sustainable SCPHA, i.e. the solution of (15). The result shows that the performances of these two optimal protocols are close. Therefore, the system is subject to little loss with respect to the social welfare by solely considering homogeneous job allocation schemes in the design.

V. EXPERIMENTS

In this section, we provide numerical results to illustrate the features of our proposed social cloud protocol design. We deploy a social cloud system for online backup as in [15] with 1000 suppliers. The arrival rate of tasks is set to be 50/min. A unit price $q = 5$ and a unit cost $c = 1$ are utilized. We also set $K_{min} = 5GB$ and $K_{max} = 25GB$, and the loads of incoming tasks follow a distribution $g(K) = \phi(\frac{K-\mu}{\sigma}) / \int_{K_{min}}^{K_{max}} \phi(\frac{K-\mu}{\sigma})dK$, where ϕ is the standard normal distribution with the mean value μ and the standard deviation σ. Several examples of $g(K)$ are illustrated in Figure 3 with $\sigma = 5$. We assume that the capacity of an individual supplier is $40GB$ and the duration of each task is 30 minutes [12]. We consider the job allocation schemes $\Gamma = \{\varphi \mid \varphi(K) = \varphi(K'),\ \text{if}\ \lfloor K / 2.5 \rfloor = \lfloor K' / 2.5 \rfloor\}$.

We first analyze the problem (12) of the optimal SCP design. Figure 4 illustrates the optimal job allocation scheme $\varphi^*(K)$. As proved in Proposition 4, $\varphi^*(K)$ is a monotonically increasing step function over $K \in [K_{min}, K_{max}]$. Hence, when the load of a task becomes larger, it is always optimal to assign to it a larger working group. Meanwhile, when the mean value μ decreases, the distribution of K becomes

[12] It should be noted that the selection of the task duration does not affect our experiments, and we select this value in order to ensure sufficient tasks take place during the experiment and generate valid results.

more concentrated around K_{min} with a small average load $\bar{K}$ per task. Most of the incoming tasks are small in this case, and hence the average arrival rate of jobs for each individual supplier also becomes smaller. This decreases a supplier's incentive to cooperate since its expected future benefit (i.e. future revenue) upon cooperation decreases. Therefore, at any value of K, $\varphi^{*}(K)$ becomes larger when μ decreases, in order to reduce the load of each job (which reduces a supplier's instant gain from deviation) and maintain its incentive to cooperate. Figure 5 illustrates the optimal incentive mechanism against the error probability ε. We only plot the optimal punishment probability β_{φ^*} since $\alpha_{\varphi^*} = 1$ always holds. For illustration purposes, we set $\beta_{\varphi^*} = 1$ when there is no sustainable SCP. A supplier's incentive to cooperate monotonically decreases when the error probability ε increases. Hence, β_{φ^*} monotonically increases against ε in order to provide sufficient punishment. Also, a similar phenomenon could be noticed when the discount factor δ increases. When a supplier puts less weight on its future utility and becomes more interested in the instant gain from the current job, larger punishment needs to be enforced.

Next, we compare the performances of the designed SCPs. Three protocols are deployed: (i) the optimal SCP which is the solution of (12); (ii) the optimal SCPHA which is the solution of (15); and (iii) a naïve SCP with $\alpha = 1$, $\beta = 1$, which is similar to the widely deployed tit-for-tat strategy (i.e. if a supplier's reputation is H when it provides resources and is B otherwise), and $\varphi(K) = 1$, $\forall K \in [K_{min}, K_{max}]$, i.e. no job allocation is employed. In the experiment, each supplier does not know the exact values of α, β and ε, and has to learn and dynamically update these values through its past experience in the system. We run the experiment for 100 hours and the resulting social welfares are plotted in Figure 6. The plotted social welfares are normalized with the social optimum $q\bar{K}$ and hence take values from $[0,1]$. It could be noted that the optimal SCP and the optimal SCPHA delivers close performances that are also near the social optimum (normalized social welfare close to 1) when the error probability ε is small and suppliers are sufficiently patient with a small discount factor δ. Hence, the optimal SCPHA can be used as a good and simple approximation to the optimal SCP in the practical system design. Meanwhile, the optimal protocols significantly outperform the naïve SCP. Therefore, the performance of a system can be remarkably improved when the SCP is carefully designed. Table 3 shows the average storage space used on an individual supplier. For example, the first column in Table 3 shows that when $\mu = 10$ and the optimal SCP is deployed, the supplier has to contribute a storage space smaller than 10GB for 17% of the time, between 10 and 20GB for 75% of the time, and between 20 and 30GB for 8% of the time. Hence, it is sufficient for each supplier to have a capacity which is at least $30GB$ to ensure a good quality of service under both the optimal SCP and SCPHA. This could be then set as the minimum capacity requirement to allow a supplier to join the social cloud system.

So far, we have adopted a random supplier selection rule in the job allocation scheme. That is, the operator randomly selects suppliers who have sufficient idle storage spaces with no preference to suppliers of high reputations. In the following experiment, we adopt a reputation-based supplier selection in the job allocation scheme and analyze whether such change improves the performance of the system. Particularly, we consider a reputation-based supplier selection rule as follows: when a supplier has sufficient idle storage space to accept a job, it has a probability ρ_H to be selected if its reputation is H and has a probability $\rho_B < \rho_H$ to be selected if its reputation is B. Figure 7 shows the system performance under different supplier selection rules when the SCP is optimally designed. The reputation-based supplier selection rule only delivers slight improvement on the system performance when the discount factor δ is sufficiently small (i.e. suppliers are sufficiently patient). Nevertheless, when suppliers become less patient with δ increasing, reputation-based supplier selection rules significantly degrade the system performance. The main reason for this phenomenon is that with a smaller ratio ρ_B / ρ_H, suppliers of reputation B expect fewer jobs and thus less benefit in the future. Therefore, their incentives to cooperate decrease with ρ_B / ρ_H and more free-ridings can be observed as shown in Figure 8, where the fraction of negatively reported tasks among all tasks is plotted [13]. It is thus optimal to deploy the random supplier selection rule in the design of SCPs.

VI. Conclusion and Future Research

In this paper, we propose a general framework to jointly design incentive mechanisms and job allocation schemes in social cloud systems. We focus on the design of optimal social cloud protocols in which no supplier can gain by free-riding and hence, the strategic suppliers find it in their self-interest to behave cooperatively. We investigated the structural results of the optimal protocols. Specifically, we proved that a simple two-level reputation scheme is sufficient in order to design optimal protocols. Moreover, we showed that a simple homogeneous allocation scheme is sufficient to provide correct incentives to suppliers and deliver a close to optimal performance. We also rigorously analyzed the relationships between our designed protocol, the intrinsic social cloud system parameters (e.g. costs, errors, suppliers' patience), and the incentives of suppliers to provide resources. Finally, we conducted illustrative simulations to verify our analysis. The obtained results indicated that our proposed protocols can significantly improve the performance of emerging social cloud systems. Our framework can be extended in various directions, among which we mention three. First, suppliers in the community do not need to be homogeneous in terms of their resource provisioning costs. Different suppliers can incur

[13] It should be noted though that in Figure 8, there are more negative reports when $\rho_B / \rho_H = 0.7$ compared to the case when $\rho_B / \rho_H = 0.5$. The reason for this phenomenon is that there are more suppliers of reputation *H* free-riding when $\rho_B / \rho_H = 0.7$, even though there are less suppliers of reputation *B* free-riding in this case.

different costs when providing their resources. Also, suppliers can have different discount factors when evaluating their long-term utilities. In this case, the proposed design will need to be amended to take the heterogeneity of suppliers into account. Second, instead of always reporting truthfully, buyers can strategically or maliciously report the suppliers' actions to the cloud operator. In such cases, the proposed design will need to be made robust against such strategic/malicious behavior of the buyers. Third, we use a belief model where each user calculates the group size based on the average task load. Alternative and more sophisticated belief models can be considered and evaluated, and their impact on the resulting design of incentive protocols needs to be assessed for various deployment scenarios.

REFERENCES

[1] A. Beloglazov, R. Buyya, Y. Lee, and A. Zomaya, "A Taxonomy and Survey of Energy-Efficient Data Centers and Cloud Computing Systems", *Advances in Computers*, 82, pp. 47 – 111, 2011.

[2] C. Teixeira, R. Azevedo, J. S. Pinto, and T. Batista, "User Provided Cloud Computing," *IEEE/ACM Int'l Conf on Cluster, Cloud and Grid Computing (CCGrid)*, 2010.

[3] K. Chard, K. Bubendorfer, S. Caton, and O. Rana "Social Cloud Computing: A Vision for Socially Motivated Resource Sharing," *IEEE Trans. on Services Computing*, 99, pp. 1 – 15, 2011.

[4] Y. T. Larosa, J. Chen, D. Deng, and H. Chao, "Mobile Cloud Computing Service Based on Heterogeneous Wireless and Mobile P2P Networks," *Int'l Wireless Commun. and Mobile Comput. Conf.*, 2011.

[5] M. Lin, A. Wierman, L. Andrew and E. Thereska, "Dynamic Right-Sizing for Power-Proportional Data Centers," *IEEE INFOCOM*, 2011.

[6] R. Urgaonkar, U. Kozat, K. Igarashi and M. Neely, "Dynamic Resource Allocation and Power Management in Virtualized Data Centers," *Network Operations and Management Symposium*, 2010.

[7] D. Kusic, J. Kephart, J. Hanson, N. Kandasamy and G. Jiang, "Power and Performance Management of Virtualized Computing Environments via Lookahead Control," *Journal of Cluster Computing*, 12(1), pp. 1 – 15, 2009.

[8] A. Gandhi, M. Harchol-Balter, R. Das and C. Lefurgy, "Optimal Power Allocation in Server Farms," *11th Int'l Joint Conf. on Measurement and Modeling of Computer Systems (SIGMETRICS)*, 2009.

[9] H. Amur, J. Cipar, V. Gupta, G. Ganger, M. Kozuch and K. Schwan, "Robust and Flexible Power-Proportional Storage," *1st ACM Symposium on Cloud Computing*, 2010.

[10] D. Warneke, O. Kao, "Exploiting Dynamic Resource Allocation for Efficient Parallel Data Processing in the Cloud," *IEEE Trans. on Parallel and Distributed Systems*, 22(6), pp. 985 – 997, 2011.

[11] K. Graffi, D. Stingl, C. Gross, H. Nguyen, A. Kovacevic, and R. Steinmetz, "Towards a P2P cloud: Reliable Resource Reservations in Unreliable P2P Systems," *IEEE Int'l Conf. on Parallel and Distributed Systems (ICPADS)*, pp. 27 – 34, 2010.

[12] L. Rao, X. Liu, M. Ilic, and J. Liu, "Distributed Coordination of Internet Data Centers under Multiregional Electricity Markets," *Proc. of IEEE*, 100(1), pp. 269 – 282, 2012.

[13] M. Mihailescu and Y. M. Teo, "On Economic and Computational-efficient Resource Pricing in Large Distributed Systems," *IEEE/ACM Int'l Conf on Cluster, Cloud and Grid Computing (CCGrid)*, pp. 838 – 843, 2010.

[14] D. Niyato, A. V. Vasilakos, and Z. Kun, "Resource and Revenue Sharing with Coalition Formation of Cloud Providers: Game Theoretic Approach," *IEEE/ACM Int'l Conf on Cluster, Cloud and Grid Computing (CCGrid)*, pp. 215 – 224, 2010.

[15] S. Seuken, D. Charles, M. Chickering, and S. Puri, "Market Design & Analysis for a P2P Backup System," *ACM Conf. on Electronic Commerce (EC)*, pp. 97 – 108, 2010.

[16] M. Feldman, C. Papadimitriou, J. Chuang, and I. Stoica, "Free-riding and Whitewashing in Peer-to-Peer Systems", *Workshop on Practice & Theory of Incentives in Networked Systems*, 2004.

[17] Y. Larosa, J. Chen, D. Deng and H. Chao, "Mobile Cloud Computing Service Based on Heterogeneous Wireless and Mobile P2P Networks," *7th Int'l Wireless Commun. and Mobile Computing Conf.*, 2011.

[18] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman, "Reputation Systems", *Communications of the ACM*, 43, pp. 45 – 48, 2000.

[19] M. Kandori, "Social Norms and Community Enforcement", *Review of Economic Studies*, 59(1), pp. 63 – 80, 1992.

[20] R. Sutton and A. Barto, "Reinforcement Learning: An Introduction", *MIT Press*, Cambridge, MA, 1998.

[21] A. Leizarowitz and A. J. Zaslavski, "Uniqueness and Stability of Optimal Policies of Finite State Markov Decision Processes", *Mathematics of Operations Research*, 32, pp. 156 – 167, 2007.

[22] Y. Zhang and M. van der Schaar, "Online Appendix: Incentive Provision and Job Allocation in Social Cloud Systems," http://medianetlab.ee.ucla.edu/papers/social_cloud_jsac_appendix.

FIGURES

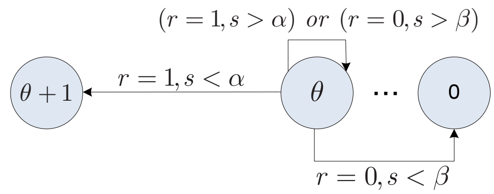
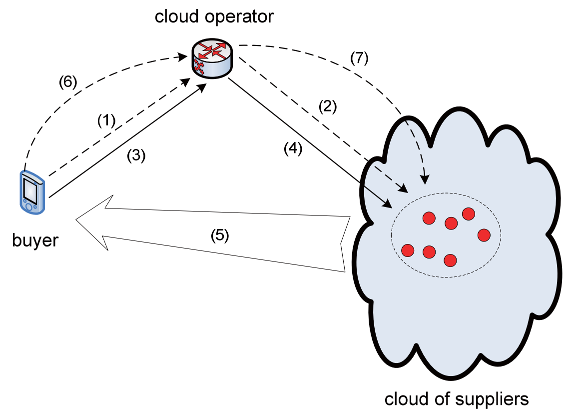


Figure 1 The schematic representation of the reputation scheme



(1) The buyer initiates a task and submits it to the operator;
(2a) The operator receives the task, specifies its load *K*, divides it with the job allocation scheme $\varphi(K)$;
(2b) The operator selects a group *G* of suppliers with sufficient idle resources;
(2c) The operator determines the total price the buyer has to pay for this task with each supplier whose reputation is no less than *h* charging $qK / \varphi(K)$ and each supplier whose reputation is below *h* charging nothing;
(3) The buyer submits the payment for this task;
(4) The operator allocates the jobs and the associated payments to the selected suppliers;
(5) The suppliers jointly provide resources to the buyer;
(6) After the task ends, the buyer reviews the task and submits its report *r*;
(7) The supplier receives the report *r* and updates the reputation of each supplier in the group with τ.

Figure 2 The operations in one task

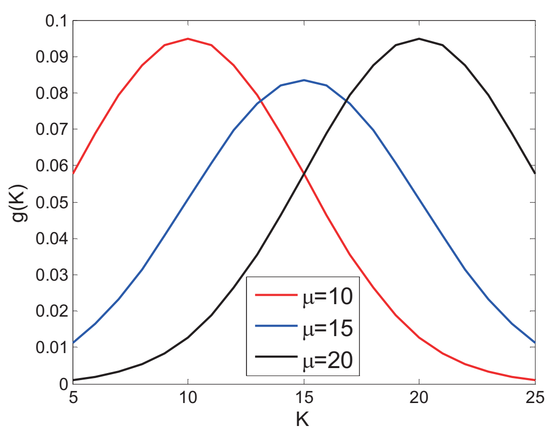


Figure 3 Examples of $g(K)$ ($\sigma=5$)

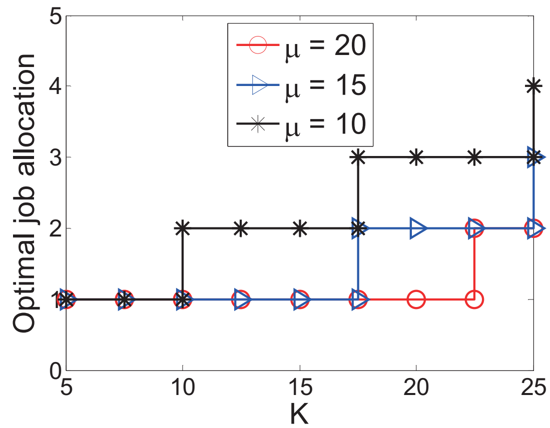


Figure 4 The optimal job allocation scheme

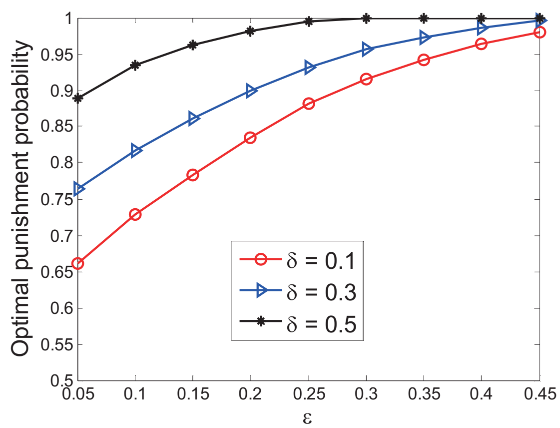


Figure 5 The optimal incentive mechanism

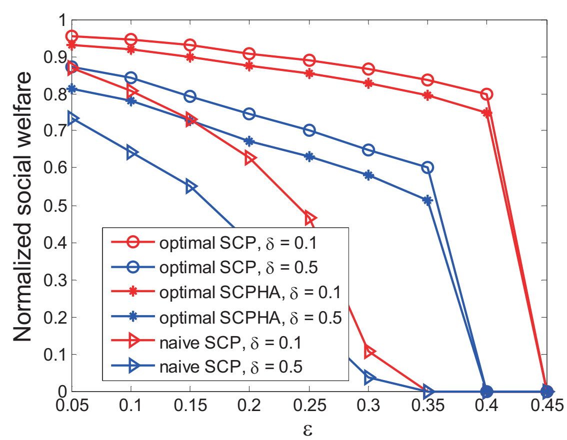


Figure 6 Performance of difference SCPs

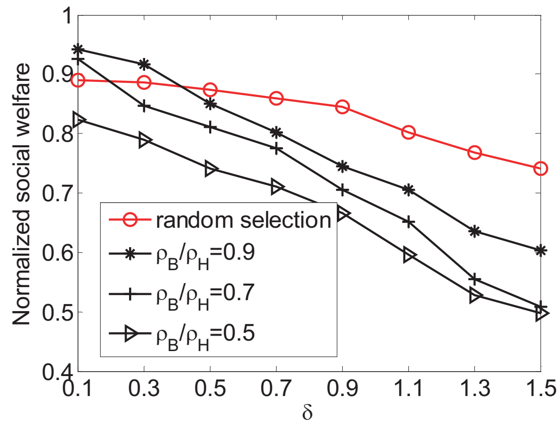


Figure 7 Performances of different supplier selection rules

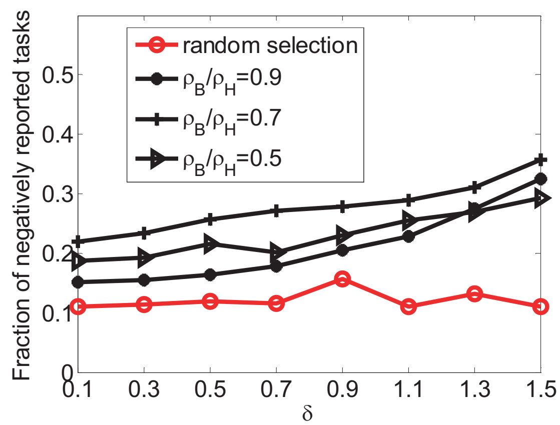


Figure 8 Negative report ratios of different supplier selection rules

Tables

Table 1. Comparison of the existing literature and our work

	[5]-[10]	[2][3][11][12]	[13]-[15]	Our work
System architecture	Server-client	Peer-to-peer	Peer-to-peer	Peer-to-peer
Resource management	Centralized	Distributed	Distributed	Distributed
Supplier characteristics	Obedient	Cooperative	Self-interested	Self-interested
Incentive design	No	No	Yes	Yes
User interaction	N/A	One-shot	One-shot	Repeated
System performance optimization	Yes	No	No	Yes

Monitoring requirement	N/A	Individual behavior of each supplier	Individual behavior of each supplier	Collective behavior of suppliers

Table 2. Summary of variables

	Notation	Meaning
	K	The load of a task
	k	The load of a job
System Setting (predetermined and fixed in the protocol design)	q	The price for unit resource
	c	The cost for the provision of unit resource
	ε	The error probability in a job
	δ	The discount factor for suppliers
	K_{min}	The minimum load of a task that can be accepted by the system
	$K_{\max}$	The maximum load of a task that can be accepted by the system
Design Parameters	L	The reputation length
	h	The cutoff threshold in the pricing scheme
	α	The forgiving probability
	β	The punishment probability
	φ	The job allocation scheme

Table 3. The average storage space usage ($\mu = 10$)

	Percentage of time that an invididual supplier spends on different usage level		
Storage usage	Optimal SCP	Optimal SCPHA	Naïve SCP
(0,10GB)	17%	13%	54%
(10GB, 20GB)	75%	76%	28%
(20GB, 30GB)	8%	11%	14%
(30GB, 40GB)	0%	0%	4%
Total	100%	100%	100%