Towards Peer-to-peer based Service Hosting through Procurement Mechanism for Host Selection

Bingqing Shen, Jingzhi Guo and C. L. Philip Chen

Faculty of Science and Technology, University of Macau,

Macau, China

{yb17412, jzguo, philipchen}@umac.mo

***Abstract*— Existing peer-to-peer systems focus on mono-type service provision, such as file sharing or file storage. If it is required to build more complex applications on peer-to-peer networks such as a virtual world application, they have to be extended to hosting diversified services. To achieve this, we propose a peer-to-peer platform on which peer computers can jointly host a file, a website, or a program. Users of our platform can utilize spare hardware resource on peer computers to host their own services. They pay the hosting cost to peers for hardware resources utilization. The problem is how to minimize hosting cost while maintain required service availability. This problem becomes difficult when peers are strategic. In this paper, we provide a procurement mechanism is provided to solve this problem by inducing peers reporting their true hosting cost and availability.**

***Keywords*: Service hosting; Procurement; Peer availability; Service availability; Cost minimization**

I. INTRODUCTION

Nowadays, computer services are more deployed on dedicated servers or data centers (e.g., cloud computing). Large spare computation resource on user machines has not been fully exploited, which is a big waste. The emergence of peer-to-peer (P2P) technology aims at addressing this issue. But existing P2P systems focus on mono-type service provision. For example, [17] only provides file sharing function that a popular file can be easily searched and retrieved. Another example is [9], P2P file storage system, that personal files stored on the system will not get lost. If we need to build more complex applications on P2P networks such as a virtual world application, they have to be extended to host diversified services. A simple example is that some users want to share files, while others need to store personal files, or even try the combination of the former two. To achieve this goal, this paper proposes a P2P platform on which peer computers can jointly host a file, a website, or a program. Users of this platform can utilize spare hardware resource on peer computers to host their own services. They pay the hosting cost to other peers for hardware resources utilization. In this system, peers are called *hosting providers*.

To design such a P2P platform, the first problem needing to solve is to find a scheme such that a service can be deployed on a set of peer computers, with which required service availability with minimum hosting cost can be achieved. *Service availability* is a probability measure of finding a workable service. Nevertheless, different services may have different availability requirements. As observed in [13], peer computers hosting different P2P applications show different up and down patterns. Each of the patterns, as we named, is *peer availability*. It is a probability measure of online status for a peer computer to host its services together with the considerations of hosting cost and profit affected by the hosting behavioural strategy. A hosting provider may cheat on its hosting cost or peer availability to maximize profit thus affecting the entire hosting cost and service availability. To fairly increase the service availability with minimum hosting cost, this paper proposes a properly designed procurement mechanism in which a hosting provider will be induced to report the hidden information of hosting cost and peer availability. In this procurement, a user becomes a *buyer* and hosting cost of an available peer for a specific service is reported by a set of qualified providers. The buyer then decides from which set of peer computers (i.e., a set of replicas) the service will be purchased after collecting the service quotations. In this procurement model, there are three particular research problems needing to be resolved.

Incentive compatibility problem. Hosting providers are rational, that is, they are self-interested to pursue maximal utility. Thus, they have incentive to both overbid hosting costs and underbid peer availability in procurement [6]. This leads to that incentives are incompatible with the overall goal of service availability.

QoS Deviation Problem Even if the first problem is solved, he may deviate from the agreed QoS in the daily operation process if he can gain more by utilizing the hardware resource for other purposes.

Fair Payment Allocation Problem. Given that a specific service is available from a group of hosting providers, what should a fair payment allocation scheme be devised to the group members?

This paper aims at solving the first problem by providing a procurement mechanism such that service cost is minimized while desired service availability is achieved by inducing hosting providers' true hosting cost and service availability distribution. To resolve this problem is important because it can guarantee the desired service availability at minimized cost. In addition, as far as we know, this problem has not been studied in P2P research field and is worth of a detailed research.

The rest of the paper is organized as follows. Section II describes the related work regarding P2P file storage and incentive models. Section III defines peer availability. Section

IV describes the problem in existing procurement mechanism. Section V proposes a properly designed procurement mechanism to solve the problem and provides an algorithm for implementation. Section VI demonstrates the correctness of the proposed procurement through an example. Finally, Section VII draws a conclusion, informs the research implications, and provides the future work.

II. Related Work

A. P2P File Storage in Structured Overlay Network

1) File Storage on PAST

P2P file storages are typically tied with the underlying P2P overlay network that is specially designed. In the case of structured P2P networks, a typical approach exploits closeness in the ID space to proactively replicate objects. One of the most popular solutions in this class is PAST [11], which is built on Pastry [12]. Besides PAST, [4] and [5] proposed approaches conceptually similar to PAST.

2) Replication Strategy for High Availability

The [3] studied the relation of minimum replica number k and file availability. The *file availability* (FA) [3] was defined as the probability that at least one host is up. By assuming each peer has the same failure distribution, the minimum number of replica to achieve a given availability level can be determined.

Contrast to FA, [16] and [7] proposed an optimal replication schemes based on file *request hit rate* (RHR). RHR is defined as the probability that a request for any file can be served in the system. Given the request rate r_j of file j and average peer failure probability q, the conceptually optimal replication number is logarithmically proportionally to r_j [7].

Maintaining high file availability in a P2P system with high utilization is only possible if the total amount of storage space in the system does not decrease. Otherwise, when beyond a certain point, the system would be unable to re-replicate files to compensate replicas' lost due to the lack of storage space in the system. This leads to the study of incentive and economics in P2P systems.

B. Payment-based Incentive in P2P Systems

P2P networks provide a platform that users can exchange resources freely, openly and anonymously. It is reasonable to model P2P networks as an economic system in which peers share public goods. However, free-riding, i.e., using other peers' resources without contributing self's resources, becomes a problem [8]. In existing P2P systems, such as Gnutella, nearly 70% users share no files while only 1% peers serve almost 50% file requests [1].

A number of incentive mechanisms based on economics and social science has been proposed to address the free-riding issue. One type is simply based on bartering. Tit-for-tat strategy adopted in [17] and [9] falls into this type. However, they do not support saving. Another type utilizes reputation and trust models. Peers maintain their own account book and keep reputation record of others learnt from trades. The problem is that reputation is not transitive and then consistency is hard to be maintained. The third type is payment-based, which utilizes currency or tokens as media to overcome the transitive problem in reputation systems. Some initial attempts to implement payment systems in P2P systems can be found in [19] and [18].

Payment-based file storage schemes provide a fairer platform for user resource exchange. Some schemes, such as Karma [15], employ auctions as a mechanism for resource trade. Nevertheless, none of the existing works studies the characteristics of trades in P2P setting in detail. Peers in an auction may overbid the hosting cost or underbid their availability. This may lead to overcharging from a user.

III. Peer Availability Distribution

The [13] states that most peers are long-lived in P2P systems, but their uptime period is limited. They appear only once per day. Replication based on FA and RHR causes large file migration overhead due to transient state. This paper believes that failure of peer availability has diurnal correlation [2] among peer computers. Thus, to avoid replica migration overhead, a service is deployed on a group of hosting providers who can jointly maintain high availability within a given period daily, which could be 24 × 7 hours (see Fig. 1), for instance.

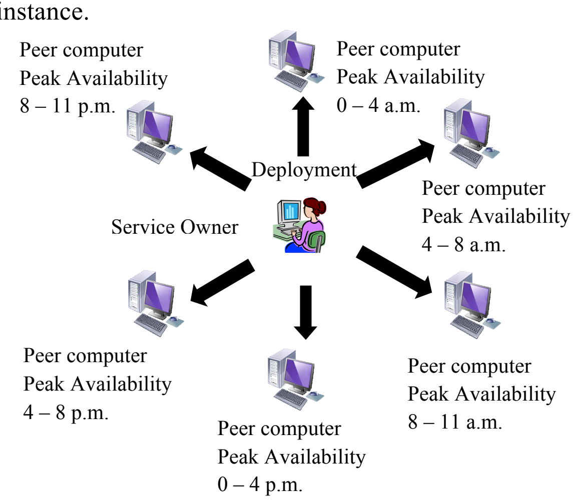

Fig. 1 Service hosting on peer computers at differnt availability distribution

As in Fig. 1, peer availability is represented by the probability (P) that peer computer is online and all the required resources for service hosting are available at time slot j. More formally, peer availability (A) at time slot j of time t for computer online status s = (0: offline, 1: online) with hosting capability requirements (q_1, q_2, …, q_n) is denoted as follows:

$$A_i^j = P_i(t = t_j, q_1 \ge \underline{q_1}, q_2 \ge \underline{q_2}, q_3 \ge \underline{q_3}, q_4 \ge \underline{q_4}, s = 1) \qquad (1)$$

in which $\underline{q_1}$, $\underline{q_2}$, …, $\underline{q_n}$ represent the minimum capacity requirements to host a specified service, for example, q_1 = CPU, q_2 = Memory, q_3 = Storage, q_4 = Bandwidth.

Based on Formula (1), *joint peer availability at time slot j* achieved by all members of a peer computer group G is:

$$A^j = 1 - \bigcup_{i \in G} (1 - A_i^j) \quad (2)$$

Likewise, based on Formula (1), *peer availability distribution of peer i over k time slots* is denoted as:

$$\boldsymbol{A}_i = \{A_i^1, A_i^2, ..., A_i^k\} \quad (3)$$

Combining Formula (2) and (3) together, the *service availability distribution* of group *G* over *k time slots* is denoted as:

$$\boldsymbol{A} = \{A^1, A^2, ..., A^k\} \quad (4)$$

Definition 1 (*domination on peer availability distribution*). For $\boldsymbol{A}_i$ and $\boldsymbol{A}_j$, if $A_i^u \geq A_j^u$ for each time slot $u = 1, 2, \ldots, k$, $\boldsymbol{A}_i$ is said to *weakly dominates* $\boldsymbol{A}_j$, denoted as $\boldsymbol{A}_i \geq \boldsymbol{A}_j$. If $\boldsymbol{A}_i \geq \boldsymbol{A}_j$ and there exists a time slot v such that $A_i^v > A_j^v$, $\boldsymbol{A}_i$ is said to *dominates* $\boldsymbol{A}_j$, denoted as $\boldsymbol{A}_i > \boldsymbol{A}_j$.

IV. Service Hosting Procurement Problem

In service hosting procurement, a user distributes his services on several peer computers. He will collect hosting providers' offers including the information of hosting cost and peer availability. After allocating his services to peer computers, he will pay the providers. This section will illustrate the research problem in this procurement process through a motivational example.

A. A Motivational Example

Suppose a user wants to run a service during 6 a.m. to 10 p.m. at GMT+08. This 14-hour period is divided to 7 slots. The user calls on a procurement with the hosting providers listed in Table 1. Two or more providers can jointly provide the hosting to fulfil the availability requirement. For simplicity, only two-provider unions are allowed. The service availability requirement is shown in Table 2. Since the user does not have Table 1, he needs a strategy to minimize the hosting cost while the availability requirement can be fulfilled.

Table 1 Peer Availability Distribution and Hosting Cost

Provider Index	***Slot 1 (A / C)***	***Slot 2 (A / C)***	***Slot 3 (A / C)***	***Slot 4 (A / C)***	***Slot 5 (A / C)***
1.	0.7 / 10	0.8 / 50	0.6 / 50	0.4 / 50	0.3 / 50
2.	0.7 / 20	0.4 / 20	0.2 / 5	0.3 / 5	0.4 / 10
3.	0.9 / 50	0.4 / 10	0.4 / 10	0.5 / 10	0.8 / 30
4.	0.5 / 10	0.7 / 20	0.9 / 40	0.1 / 50	0.0 / 0
5.	0.3 / 5	0.4 / 30	0.6 / 30	0.7 / 30	0.5 / 30
6.	0.2 / 20	0.5 / 40	0.6 / 40	0.9 / 50	0.6 / 40
7.	0.1 / 10	0.3 / 20	0.4 / 20	0.7 / 40	0.7 / 30

A: Peer availability
C: Hosting cost

Table 2 Availability Requirement

Slot 1	***Slot 2***	***Slot 3***	***Slot 4***	***Slot 5***
0.7	0.8	0.7	0.8	0.7

In the case of slot 1, provider 1 may mistakenly believe that provider 2's value is 30. Provider 1 therefore bids 29 and if provider 2 bids any amount smaller than this value he will win the item. Thus, a procedure is needed that will select the hosting provider with the highest value regardless of the accuracy of the beliefs of the participants.

The common approach is to use Vickrey auction [14] that the winner(s) will be paid with the price of the first loser. It has been proved [14] this approach can induce providers to report their true prices of a good. In this example, provider 1 wins the auction and gets paid 20 from provider who is the first loser.

However, if provider 1 colludes with provider 2 and they both falsely report the availability equal to 0.6, then the winner is (provider 1 + provider 2) and the loser is provider 3. The winner will get paid 50 which is higher than their honest payment. In this case, apparently the user is overcharged.

In summary, without the knowledge of providers' hosting cost and actual peer availability distribution, the user could be overcharged by the providers even if the service availability requirement is fulfilled. The problem in this paper is to design a proper procurement mechanism so that truly reporting hosting cost and peer availability is the best strategy for all providers. The rest of this section formally defines buyer's objective and seller's objective respectively, and then converts the procurement mechanism design to an optimization problem.

B. Buyer's Objective

A user wants to deploy a file or a service to several hosting providers which can jointly meet the availability requirement. For incentive purpose, service hosting is not free so that the buyer also wants to minimize the hosting cost. This becomes an optimization problem that the buyer's objective is to minimize the expected cost with a fixed availability requirement, expressed as

$$\text{Minimize } \mathrm{E}(\sum_i \sum_j t_i^j) \text{ for each j}$$

$$\text{subject to } \boldsymbol{A} \geq \boldsymbol{A}_r \text{, where } \boldsymbol{A}_r \text{ : required availability)}$$

where t_i^j is the expected payment transfer from the buyer to hosting provider i at time slot j (implying that the cost of hosting is also correlated to time) and $\boldsymbol{A}_r$ is the required availability.

C. Seller's Objective

Each service provider has the hosting cost $c_i^j \in [\underline{c}, \overline{c}]$ at time slot j. Provider i may hide the true value c_i^j and report the fake cost $\hat{c}_i^j$ to maximize her income (usually $\hat{c}_i^j \geq c_i^j$). It is assumed that the lower bound ($\underline{c}$) and higher bound ($\overline{c}$) of cost is public to all participants. They can be learned from estimation and history procurements. Moreover, cost at different time slots are independent. Similarly, a hosting provider may also report a peer availability ($\hat{A}_i^j$) differing from the true value (A_i^j), where $\hat{A}_i^j, A_i^j \in [\underline{A}, \overline{A}]$ in a range public to all participants.

Let $x_i^j \in \{1, 0\}$ denote the *allocation function* representing the whether provider i wins the bid for hosting at time slot j.

The provider's offered surplus (ρ_i^j , observed by the buyer) [6] at time slot j is denoted as

$$\rho_i^j = E[t_i^j(\hat{c}_i^j, \hat{A}_i^j) - \hat{c}_i^j x_i^j(\hat{c}_i^j, \hat{A}_i^j)] \quad (5)$$

and the expected surplus (π_i^j) is

$$\pi_i^j = E[t_i^j(\hat{c}_i^j, \hat{A}_i^j) - c_i^j x_i^j(\hat{c}_i^j, \hat{A}_i^j)] \quad (6)$$

π_i^j must be greater than or equal to zero, otherwise providers will not participate the procurement. Hosting provider's goal is to maximize his total expected surplus $\pi_i = \sum_j \pi_i^j$, denoted as $Max(\pi_i) = Max(\sum_j \pi_i^j) = \sum_j Max(\pi_i^j)$ since the procurements in each time slot are independent.

D. Problem Consolidation

Combining buyer's objective and hosting providers' goal together, the problem of designing a proper procurement mechanism can be converted to solving an optimization problem with the following objective function and constraints.

$$\begin{aligned} &\text{Minmize } \mathrm{E}[\sum_i \sum_j t_i^j] \text{ for each j} \\ &\text{subject to } \boldsymbol{A} \geq \boldsymbol{A}_r \\ &\text{and } (c_i^j, A_i^j) \in \underset{\substack{c_i^j \in [\underline{c}, \bar{c}] \\ A_i^j \in [\underline{A}, \bar{A}]}}{argmax}(\pi_i^j) \\ &\text{and } \pi_i^j \geq 0 \end{aligned} \quad (7)$$

In (7), the first constraint regulates the minimum peer availability requirement. The second constraint is called *incentive compatibility* (**IC**) constraint. This constraint requires that reporting true value of hosting cost and availability can maximize providers' surplus. If a procurement mechanism meets the IC constraint, our problem can be solved. The last constraint is called *individual rationality* (**IR**) that hosting providers do not pay for entrance.

V. Procurement Mechanism

Optimal mechanism design was first proposed by Myerson whose work [10] laid a solid foundation in this area. Iyengar and Kumar [6] further developed the optimal mechanism design theory and applied it in procurement of divisible goods. We adapted their work to P2P service hosting procurement. In this section, the procurement mechanism for service hosting is introduced first. Then, the procurement mechanism is characterised for the specific optimization problem.

A. The Procurement Process

It is an one-round procurement to avoid providers learning from history and adopting favourable strategies. The process is shown in Fig. 2 and it can be divided into three stages.

- Stage 1, the buyer sends an inquiry to all sellers. This can be achieved by broadcasting the inquiry message in the network. How to design an efficient broadcast protocol in a P2P network is another research problem and it will not be discussed in this paper.
- Stage 2, all qualified sellers (or sell groups) send back information of their hosting cost and peer availability distribution (or joint peer availability distribution), on receiving buyer's inquiry.
- Stage 3, eventually, the buyer selects the seller (or the sell group) who can provider the required service hosting with minimum charge.

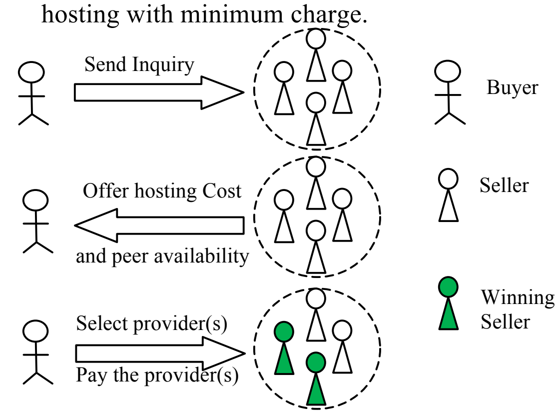

Fig. 2 The procurement process

B. Characterising the Procurement Mechanism to the Optimization Problem

Suppose we have a procurement mechanism $\mathcal{M}(\boldsymbol{t}, \boldsymbol{x})$ which can solve (7). $\boldsymbol{t}$ represents the payment vector and $\boldsymbol{x}$ represents the allocation function vector. To solve the optimization problem, $\mathcal{M}$ must be characterised in $\boldsymbol{t}$ and $\boldsymbol{x}$ to meet the objective function and all constraints. It is done on the **IC** constraint first.

Lemma 1 A mechanism $\mathcal{M}(\boldsymbol{t}, \boldsymbol{x})$ is incentive compatible (**IC**) if the allocation $x_i^j(u, A_i^j)$ is non-increasing in hosting cost c_i^j and non-decreasing in availability A_i^j for any provider i at time slot j. The proof of *Lemma 1* is similar as in [6].

Buyer's objective is to minimize the expected hosting cost. Since the costs between any two time slots are independent, then $\mathrm{E}(\sum_i \sum_j t_i^j) = \sum_j \mathrm{E}(\sum_i t_i^j) = \sum_j \mathrm{E}^j(\sum_i t_i)$ so that the procurement at each time slot can be designed independently. Let $f(c_i^j)$ and $F(c_i^j)$ denote the probability density function and the cumulative density function of choosing cost c_i^j . With *Lemma 1*, the following theorem characterises the objective function and the payment scheme.

Theorem 1 If $\mathcal{M}(\boldsymbol{t}, \boldsymbol{x})$ is incentive compatible (**IC**), the objective function in (5) is equivalent to minimize

$$E^j[\sum_i x_i(c_i^j, A_i^j)(c_i + \frac{F(c_i^j)}{f(c_i^j)})] \quad (8)$$

And the payment function is

$$t_i^j(c_i^j) = c_i^j x_i^j(c_i^j) + \int_{c_i}^{\bar{c}} x_i^j(u)d(u) \quad (9)$$

The proof of *Theorem 1* needs *Lemma 1* and the proving process is similar as in [6].

We use H_i^j denote the *virtual cost* defined as

$$H_i^j(c_i^j) = c_i^j + \frac{F_i^j(c_i^j)}{f_i^j(c_i^j)} \tag{10}$$

Then (8) can be converted to $E^j[\sum_i x_i(c_i^j, A_i^j)H_i^j(c_i^j)]$.

For the case that two or more providers jointly provide the hosting service to achieve the required availability, the joint virtual cost $H_{i,g}^j$ for each provider group i with size g can be calculated with

$$\begin{aligned} &H_{i,g}^j(c_i^j) = c_{i,g}^j + \frac{F_{ig}^j(c_{i,g}^j)}{f_{i,g}^j(c_{i,g}^j)} \\ &\text{where } f_{i,g}^j(c_{i,g}^j) = f_1^j(c_1^j)f_2^j(c_2^j)...f_g^j(c_g^j) \\ &\text{and } c_{i,g}^j = c_1^j + c_2^j + \ldots + c_g^j \\ &\text{and } F_{i,g}^j(c_{i,g}^j) = \int_{\underline{c}}^{c_{i,g}^j} f_{i,g}^j(u)du \end{aligned} \tag{11}$$

C. Allocation Algorithm Design

Assumption 1 (*regularity Assumption* [10]). Assume that for all n participants at k time slots, the virtual cost H_i^j is non-decreasing in c_i^j.

To design an allocation algorithm, inspecting all availability combinations to search the required one with minimum cost is NP-hard. Here a heuristic approach (see *Allocation 1*) is introduced at one time slot. Suppose at most $\mathcal{G}$ hosts are allowed at each time slot, and $\boldsymbol{A_r}$ can always be achieved by a group of providers with group size smaller than or equal to $\mathcal{G}$. This can be satisfied by narrowing the span of each time slot.

Allocation 1. Searching the minimum cost at time slot j
(*a*) Sort H_i^j in ascending order so that $H_1^j < H_2^j < \ldots < H_n^j$
(*b*) Looking for the first index *i* that $A_i^j \geq A_r^j$ and record $H_*^j = H_i^j$
(*c*) Set index $g = 2$
(*d*) Remove the first *i* biddings and calculate joint peer availabilities of any *r* providers $A_{1,g}^j, A_{2,g}^j, \ldots, A_{k,g}^j$ (with equation (1)) and joint virtual costs $H_{1,g}^j, H_{2,g}^j, \ldots, H_{k,g}^j$
(*e*) Sort $H_{i,g}^j$ in ascending order so that $H_{1,g}^j < H_{2,g}^j < \ldots < H_{k,g}^j$
(*f*) Looking for the first index *i* such that $A_{i,g}^j \geq A_r^j$ and record $H_{*,g}^j = H_{i,g}^j$
(*g*) Increment *r* by $r = r+1$
(*h*) If $r < \mathcal{G}$, go to step (*c*)
(*i*) From the local minimum cost list $\{H_*^j, H_{*,1}^j, H_{*,2}^j, \ldots, H_{*,\mathcal{G}}^j\}$, return the global minimum virtual cost

Actually if the span of each time slot is narrowed down, most single peer can achieve the required availability and the combination iteration will be reduced. This reduction can go to the extreme that all single providers are qualified. Then this allocation algorithm will become $O(nlog(n))$ with heapsort and step (c) to (h) can be skipped.

D. Characterising the Allocation Algorithm

Allocation 1 can be characterised with the following lemma.

Lemma 2 Let $\boldsymbol{x}^{*j} = \{x_1^{*j}, x_2^{*j}, \ldots\}$ represent the allocation from *Allocation 1* (i.e., the solution of problem (7)). Suppose the *assumption 1* holds. Then,

a) x_i^{*j} is non-increasing in c_i^j for all fixed A_i^j and bids from other participants.

b) x_i^{*j} is non-decreasing in A_i^j for all fixed c_i^j and bids from other participants.

Sketch of Proof If A_i^j and other providers' bids are fixed, provider i reporting high value of cost will not increase the chance to win the bid, since x_i^{*j} is sorted in virtual cost H_{ig}^j in *Allocation 1*, and H_{ig}^j is non-decreasing in c_i^j according to the *regularity assumption*. Hence x_i^{*j} is non-increasing in c_i^j. Moreover, the joint peer availability from (1) is non-decreasing in A_i^j when other group members' availability are fixed. Thereafter, similar argument proves *b*).

E. Mechanism Design Result

With *Lemma 2*, the main result of the procurement design can be claimed in the following theorem.

Theorem 2 The procurement mechanism designed in *Allocation 1* and (9) gives the optimal procurement to the problem (7).

Sketch of Proof Firstly, with *Allocation 1*, the minimum availability requirement (constraint (1)) is apparently met. Moreover, the allocation designed in *Allocation 1* claims *Lemma 2* which further claims *Lemma 1*. Hence the **IC** constrain is met. Thirdly, the provider's expected surplus can be calculated by (6) and (9). It yields $\pi_i^j = \int_{\hat{c}_i^j}^{\bar{c}} x_i^{*j}(u, \hat{A}_i^j)du$ which is non-negative (shown in the payment scheme below) and the **IR** constraint is met. Lastly, *Allocation 1* itself is the search for minimum hosting cost, fulfilling the objective function. Hence, *Theorem 2* is correct.

Theorem 2 implies that sellers are free to participate in the service hosting procurement, but their best strategy in the game is to report true value of their hosting cost and peer availability distribution (or joint peer availability distribution). This is because with this strategy, sellers' profit (i.e., the expected surplus) can be maximized. Since we have assumed that sellers in this procurement game are rational players, the have to adopt this strategy of truth tell. With true value of hosting cost and peer availability distribution, it will be very easy to pick the qualified hosting provider who charges minimum. This can be achieved by sorting all qualified providers (including provider groups) by their hosting cost in ascending order and return the first one. Thus, it solves the problem in this paper.

The next step is to calculate the payment as expressed in (9) but with t_i^j , x_1^j replaced by x_1^{*j} , t_i^{*j} respectively. the item $\int_{\hat{c}_i^j}^{\bar{c}} x_i^{*j}(u, \hat{A}_i^j)du$ can be converted to a more intuitive way. Assume provider i (or group i) wins the order of service hosting. Let $Z_i(c_{-i})$ be the supremum of all the hosting costs that provider i (or group i) can win the bid. That is,

$$z_i^j(c_{-i}^j) = \sup\{c_i^j \mid \mathrm{H}_i^j(c_i^j) \le \mathrm{H}_k^j(c_k^j), \forall \mathrm{k} \ne \mathrm{j}\} \tag{12}$$

Then, $x_i^{*j}(\hat{c}_i^j, \hat{A}_i^j)$ can be represented with

$$x_i^{*j}(\hat{c}_i^j, \hat{A}_i^j) = \begin{cases} 0 \text{ if } \hat{c}_i^j > z_i^j(c_{-i}^j) \\ 1 \text{ if } \hat{c}_i^j \le z_i^j(c_{-i}^j) \end{cases} \tag{13}$$

This gives us

$$\int_{\hat{c}_i^j}^{\bar{c}} x_i^{*j}(u, \hat{A}_i^j)du = \begin{cases} 0 & \text{if } u > z_i^j(c_{-i}^j) \\ z_i^j(c_{-i}^j) - \hat{c}_i^j & \text{if } u \le z_i^j(c_{-i}^j) \end{cases} \tag{14}$$

Finally, the payment in the procurement mechanism is calculated with

$$t_i^{*j}(\hat{c}_i^j, \hat{A}_i^j) = \begin{cases} 0 & \text{if } X_i^{*j}(\hat{c}_i^j, \hat{A}_i^j) = 0 \\ z_i^j(c_{-i}^j) & \text{if } X_i^{*j}(\hat{c}_i^j, \hat{A}_i^j) = 1 \end{cases} \tag{15}$$

Hosting provider i (or group g) is only paid when the provider (or the group) wins the bid, and paid with the maximum amount that the provider (or the group) can report to win the bid. This payment scheme is the reverse of modified Vickrey auction [10].

VI. Applying to the Motivational Example

The designed procurement mechanism can be applied to the motivational example introduced in the front. It will show that collusion and reporting false peer availability will no longer be the best strategy for a seller in the procurement. For simplicity, suppose that the hosting cost is uniformly distributed from 0 to 100, i.e., $f(c_i^j) = 1/100$ for $\forall c_i^j \in [0,100]$. Then $F(c_i^j) = c_i^j / 100$. For illustration, the allocation ($\boldsymbol{x}$) and payment ($\boldsymbol{t}$) are calculated for time slot 1 only. Following *Allocation 1*, the virtual cost for single provider is

$$H_1^1 = 40, H_2^1 = 80, H_3^1 = 100, H_4^1 = 20, H_5^1 = 60, H_6^1 = 40, H_7^1 = 20$$

The hosting provider is selected which offers the minimum virtual cost and $A_i^1 \ge A_r^1 = 0.7$, which is peer 1. Then the qualified providers are removed and then only provider 4, 5, 6, 7 are left. We group any two of them together and calculate the joint peer availability and virtual cost respectively.

$$A_{4,5}^1 = 0.65, A_{4,6}^1 = 0.6, A_{4,7}^1 = 0.55, A_{5,6}^1 = 0.52, A_{5,7}^1 = 0.46, A_{6,7}^1 = 0.28,$$

$$H_{4,5}^1 = 30, H_{4,6}^1 = 60, H_{4,7}^1 = 40, H_{5,6}^1 = 50, H_{5,7}^1 = 30, H_{6,7}^1 = 60,$$

No 2-provider group is picked since none of the joint peer availability fulfils the requirement. As regulated in the example, only two-provider unions are allowed. Thus the iteration stops here and the global winner is provider 1.

Since the proposed payment strategy is the reverse of modified Vickrey auction [10], this procurement mechanism inherits all the merit of Vickrey auction [14]. Thus as in the Vickrey auction, reporting true value of host cost is the best strategy for all providers. We then check the case that provider 1 colludes with provider 2 and both falsely report their availability equal to 0.6. The joint availability is 0.84 which fulfil the requirement. However, their joint virtual cost $H_{1,2}^1 = 60$ which is higher than H_3^1. It means they will lose the bid and the winner will become provider 3. To prevent this happen, provider 1 has to report the true values of availability and cost. Thus, the problem previously shown in the motivational example is solved.

VII. Conclusion

The proposed procurement mechanism solves the problem that a user wants to host a service with minimum hosting cost and required service availability, while the hosting providers are strategic when they offer quotations. This is achieved by carefully designing the allocation and payment scheme to induce providers reporting their true value of hosting cost and peer availability, since our procurement mechanism satisfies the *incentive compatibility* constraint that providers' profit will be maximized only when they report the true values of hosting cost and availability distribution. Moreover, due to the *individual rationality* constraint, providers do not need to pay for entrance.

These achievements implicates that both users and hosting providers would like to participate, because their utilities can be maximized. (Here, user's utility is the negative hosting cost and hosting provider's utility is the profit.) This work is an important step to build a peer-to-peer platform and with which users can deploy services and applications but not use dedicated servers for operation cost reduction.

In future work, we will explore the QoS deviation problem to prevent hosting provider from deviating the contract in daily hosting, and the fair payment allocation problem. The second problem requires a way to fairly allocate payments in a group of hosting providers, such as how to divide the payment 20 between peer 4 and peer 5 in our illustrative example.

Acknowledgment

This research is partially supported by the University of Macau Research Grant No. MYRG156(Y2-L2)-FST11-GJZ.

REFERENCES

[1] E. Adar and B.A. Huberman, "Free riding on Gnutella," First Monday, vol. 5, 2000.

[2] R. Bhagwan, S. Savage and G.M. Voelker, "Understanding availability," in Peer-to-Peer Systems II, Springer, 2003, pp. 256-267.

[3] R. Bhagwan, S. Savage and G.M. Voelker, Replication strategies for highly available peer-to-peer storage systems, Department of Computer Science and Engineering, University of California, San Diego, 2002.

[4] F. Dabek, M.F. Kaashoek, D. Karger, R. Morris and I. Stoica, "Wide-area cooperative storage with CFS," ACM SIGOPS Operating Systems Review, vol. 35, pp. 202-215, 2001.

[5] K. Huang, T. Huang and J.C. Chou, "LessLog: A logless file replication algorithm for peer-to-peer distributed systems," in Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International, pp. 82, 2004.

[6] G. Iyengar and A. Kumar, "Optimal procurement mechanisms for divisible goods with capacitated suppliers," Review of Economic Design, vol. 12, pp. 129-154, 2008.

[7] J. Kangasharju, K.W. Ross and D.A. Turner, "Optimizing file availability in peer-to-peer content distribution," in INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE, pp. 1973-1981, 2007.

[8] R. Krishnan, M. Smith and R. Telang, "The economics of peer-to-peer networks," JITTA, vol. 5, pp. 31-44, 2004.

[9] M. Lillibridge, S. Elnikety, A. Birrell, M. Burrows and M. Isard, "A cooperative internet backup scheme," in Proceedings of the annual conference on USENIX Annual Technical Conference, pp. 3-3, 2003.

[10] R.B. Myerson, "Optimal auction design," Mathematics of Operations Research, vol. 6, pp. 58-73, 1981.

[11] A. Rowstron and P. Druschel, "Storage management and caching in PAST, a large-scale, persistent peer-to-peer storage utility," in ACM SIGOPS Operating Systems Review, pp. 188-201, 2001.

[12] A. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems," in Middleware 2001, pp. 329-350, 2001.

[13] D. Stutzbach and R. Rejaie, "Understanding churn in peer-to-peer networks," in Proceedings of the 6th ACM SIGCOMM conference on Internet measurement, pp. 189-202, 2006.

[14] W. Vickrey, "Counterspeculation, auctions, and competitive sealed tenders." The Journal of finance, vol. 16, issue 1, pp. 8-37, 1961.

[15] V. Vishnumurthy, S. Chandrakumar and E.G. Sirer, "Karma: A secure economic framework for peer-to-peer resource sharing," in Workshop on Economics of Peer-to-Peer Systems, 2003.

[16] Q. Xin, T. Schwarz and E.L. Miller, "Availability in global peer-to-peer storage systems," Distributed Data and Structures, Proceedings in Informatics, 2004.

[17] (2014) BitTorrent. [Online]. Available: http://www.bittorrent.com

[18] (2011) Flud Backup. [Online]. Available: http://www.flud.org/wiki/Flud_Backup

[19] (2014) MojoNation Technology Overview. [Online]. Available: http://web.archive.org/web/20011216035011/http://mojonation.net/docs/technical_overview.shtml