An Approach to Vickrey-based Resource Allocation in the Presence of Monopolistic Sellers

Hai Nam Pham[1], Yong Meng Teo[1], Nam Thoai[2] and Nguyen Tuan Anh[2]

[1]Department of Computer Science, National University of Singapore
Computing 1, Law Link, Singapore 117590
[2]Faculty of Computer Science & Engineering
University of Technology, Ho Chi Minh City, Vietnam
Email: teoym@comp.nus.edu.sg

Abstract

Market-based approaches proposed recently proved to be promising for competitive resource sharing in peer-to-peer and grid computing. Many approaches leverage on the Vickrey-Clarke-Groves (VCG) mechanism to achieve incentive compatibility which embraces truthful bidding of participating agents. This paper addresses a deficiency of VCG that to the best of our knowledge has not been studied. When one or more agents possess a large portion of the market share of resource, a monopoly situation arises. Applying VCG mechanism does not lead to an allocation because the second price cannot be mathematically determined. Using both theoretical and simulation analysis, we show the importance of addressing this problem. Our results show that monopoly situation arises in many types of market settings, from auction to exchange, and with a relatively high occurrence rate. To address this, we propose a new pricing method suitable for many market settings that achieve budget balanced and economic efficiency but relax the strategy proof property.

Keywords: VCG, incentive compatible, monopoly, mechanism design

1 Introduction

Resource sharing has grown from small number of machines within a single administrative domain to large number of machines connected across multiple organizations with different sharing policies such as in peer-to-peer and grid computing. Collaborative models where every agent is assumed to run the same protocol without changes are less suitable in competitive environments when agents aim to maximize its own interests. Several models have emerged including commodity market, proportional share (Chun & Culler 2000), bargaining (Laffont, Stockinger, Giddy & Abramson 2001), auction (Dramitinos, Stamoulis & Courcoubetis 2005, Yin & Wang 2006, Laffont et al. 2001) and exchange (Yin & Wang 2006, Kothari, Sandholm & Suri 2004, Parkes, Kalagnanam & Eso 2001, Parkes, Cavallo, Elprin, Juda, Lahaie, Lubin, Michael, Shneidman & Sultan 2005). The shared resources include compute cycles, storage space, network bandwidth, sensors or software licenses.

Competitive market-based models that depend on the reported values from the participants, such as auction and exchange, often aim to attain a property called strategy proof or incentive compatibility (Dash, Jennings & Parkes 2003). Incentive compatibility encourages the participants not to deviate from the dominant strategy and express the correct internal preference values during the allocation round. Most models use a modified version of the well known Vickrey-Clarke-Grove mechanism (VCG) (Nisan & Ronen 2001), to achieve strategy proof. When using VCG, the allocation is both strategy proof and economic efficient (Dash et al. 2003). Despite its advantages, VCG has several drawbacks. An allocation made by VCG is often not balanced in terms of the amount of money exchanged. In that case, either a surplus or a deficit will happen after the market clears and cause the market go down in a long operation. As a result, most of the works raise budget balance as the first constraint and approximate the others. Another drawback is its computational complexity, computing the payment for VCG is expensive and often NP-hard (Nisan 2000).

In our study of VCG mechanism, we observe that for models derived from VCG there are instances of monopoly situations that result in non-allocation of resources. Studies so far has ignored the treatment of such instances. Some allocations are naturally feasible, such as those with very high value bids or very low offered price, but allocation still cannot be made in the presence of monopolistic sellers. This monopoly situation happens when one or several sellers gain exclusive price control of their resources when they hold a large portion of resources in the market. The precise definitions must be defined with respect to a model, are presented in section 4. This situation occurs quite frequently in Internet-based systems, especially in the transient states, i.e., before the size of the system reaches steady state. In systems where the system size fluctuates during its life span, the monopoly situation can occur any time. Because of the transient nature of this state, the market can be restored to stable state if the allocation can be made. Such allocations also help to keep resource buyers in the market after their requests are continuously rejected and regardless of how much they increase their bids.

The contributions of our work are as follow. We provided a theoretical analysis of the monopoly situation for three market-based models, namely, auction, combinatorial auction and exchange to demonstrate that VCG cannot make allocation under such conditions. We propose a scheme that addresses this problem. Using simulation analysis, we quantified the occurrence rate of monopoly situation in the combinatorial exchange model.

The rest of the paper is structured as follows. Section 2 reviews related works. Section 3 describes the VCG mechanism and the terms used in this paper. In section 4, we present theoretical analysis for three popular market-based models to show that

the monopoly condition exist. We propose a pricing scheme for resource allocation under the monopoly condition. In section 5, we quantify the occurrence of the monopoly condition under different market settings. Section 6 contains the conclusion and future works.

2 Related Works

The VCG mechanism is used in many areas in computer science. In (Dramitinos et al. 2005), a series of short-time Vickrey auctions are conducted to bid for one second of network access. In (Takahashi & Tanaka 2003), network bandwidth is auctioned for leasing line, reselling line and allocation of bandwidth to individual users. Other uses of VCG are Grid computing (Kothari et al. 2004, Bubendorfer 2006) and sharing of peripheral devices (Rogers, Dash, Jennings, Reece & Roberts 2006).

Current models range from auction, combinatorial auction to double auction (also called exchange). Auction is popular with simple types of resource such as bandwidth or sensors (Dramitinos et al. 2005, Takahashi & Tanaka 2003, Rogers et al. 2006). The most popular form of auction is multi-unit auction which is the focus of this work. Combinatorial auction extends auction by allowing the allocation of a subset of requested resources (Bubendorfer 2006, Neumann, Schnizler, Weber & Weinhardt 2007). The VCG mechanism works well with complex bidding languages at the expense of computational complexity. It is capable of supporting advanced reservation and multiple bidding plans (Parkes et al. 2001, Bubendorfer 2006). Such complexity is relaxed by approximations that minimize the distance between a budget balanced solution and an incentive compatible solution from VCG mechanism (Parkes et al. 2001, Neumann et al. 2007, Lavi & Swamy 2005), often through linear programming (Lavi & Swamy 2005). Such approximations are still dependent on the solution found by the VCG mechanism, thus are also affected by the monopoly condition. Exchange is a variant of combinatorial auction with multiple sellers and uses VCG (Kothari et al. 2004, Yin & Wang 2006, Parkes et al. 2005). The extent of VCG mechanism includes mixed environment with both competitive and collaborative entities (Braynov & Sandholm 2003).

This paper describes an algorithm that addresses the monopoly situation in mechanisms derived from VCG. Our method is relevant to three popular trading models: auction, combinatorial auction and exchange, and can be included to such systems without modifying the architecture, algorithm or data structures. As our scheme performs allocations in the monopoly situation, it increases the rate of successful transactions and improves the overall performance of the market.

3 Review of VCG Mechanism

We introduce mechanism design and VCG mechanism which are referred in this report. A mechanism $m = (o, p)$ consists of two parts, an outcome o and a payment p for successful participants (or agents) in the mechanism (Nisan & Ronen 2001). The outcome o is a function which determines the winner among all participants. The payment p is a function describing the payment for each winner. The mechanism m is a pair of o and p computed based on inputs from all agents. The computation is as follow:

- Firstly, each agent picks a strategy a_i from the strategy pool A_i defined by the mechanism for that agent.
- Secondly, after all strategies of all agents $\{a_1, a_2, .., a_n\}$ are received, the output function $o(a_1 a_2 .. a_n)$ is computed.
- Finally, the payment function $p_i = p_i(a_1, a_2, .., a_n)$ is given to each agent, along with o.

For models leveraging on mechanism design, two main problems have to be addressed:

- Selfish behavior of agent i: Each agent has a real valuation defined by the function $v_i(o)$. For instance, in an allocation problem, v_i is the value of the allocated resources to the agent (or 0 if there is no resource allocated to that agent). On the fact that o is derived from a_i which is submitted to the mechanism by the agent i, it is possible for that agent to manipulate the outcome o. The ultimate purpose for this manipulation is to maximize his utility value, $U_i = p_i + v_i$. This selfish behavior introduces additional overheads for every agent, computational cost for selecting the deviated strategy and communication cost to get extra information. It also reduces the efficiency of the system by giving unfair benefits to some agents. To deal with this problem, the mechanism must be designed in such a way that the dominant strategy for every agent is to truthfully report the real valuation (Nisan & Ronen 2001). Such mechanisms are called truthful mechanism.
- The optimization of output function: This is also described as the winner determination problem by other authors (Parkes et al. 2001, Neumann et al. 2007, Schnizler & Neumann 2007). From a set of feasible outputs F, the mechanism maximizes (or minimizes) the winner determination function $g(o)$. The purpose of this function is to optimize some system-wide parameters. Due to this optimization, there is a trade-off between bidding flexibility and system complexity. On a comprehensive system like combinatorial auction (Nisan 2000, Neumann et al. 2007) users have a flexible language to express their bids and offers to the system, but with a higher cost to compute the outcome. In VCG mechanism, the winner determination function is also called from within the payment function repeatedly, thus making the trade-off more severe.

To address the first problem of mechanism design, many previous works use the VCG mechanism to achieve strategy proof and allocation efficiency which yields the highest transaction value. Budget balanced is often not achieved by VCG and must be done by approximation or sacrifying either strategy proof or allocation efficiency.

Definitions

Utilitarian function: A winner determination function is called utilitarian if it has the form: $g(o) = \sum_i a_i(o)$.

VCG mechanism: A mechanism belongs to VCG family if it use an utilitarian objective function and:

1. $o = argmax_o(\sum_i a_i(o))$
2. $p_i = \sum_{j \neq i} a_j(o) + h_i(a_{-i})$

where $h_i(a_{-i})$ is an arbitrary function of all real valuation of all agents except agent i. The VCG mechanism is proved to be truthful (Nisan & Ronen

2001). Furthermore, VCG is the only truthful implementation for problems with utilitarian objective function(Green & Laffont 1977).

Myerson-Satterthwaite (Myerson & Satterthwaite 1983) shows that the three properties of incentive compatibility, allocative efficiency and budget balanced cannot be achieved simultaneously. Due to this impossibility theorem, most works resort to a trade-off such as achieving budget balanced and strategy proof while trying to approximate allocation efficiency. Our work trades incentive compatibility for budget balanced and allocation efficiency, which are more important in monopoly condition where lack of competitiveness reduces the benefits of strategy proof.

4 Theoretical Analysis

This section covers the design of increasingly complex trading models from auction to exchange. We include auction, combinatorial auction and exchange models because they are not completely the subset of each other. Furthermore, the more complex model comes with additional computation cost and needs different levels of optimizations which we describe for each model design. Different levels of model complexities are also suited for different requirements of systems and should be distinguished for easier reference.

We start with the simplest model and provide a proof to show that monopoly situation exist. For others, we map similar attributes and reuse the proof. In each model, we analyze the effects and occurrence of the monopoly situation. Firstly we design the model. Secondly, we define the monopoly occurrence condition. Finally, under that condition, we show that allocation is not possible.

4.1 The Reverse Auction Model

Auctions have broad application in computer science, notably in load balancing and resource allocation. An auction consists of a single resource item to sell, multiple participants competing for that item, and an auctioneer to drive the auction operation. Auctions exist in a variety of forms and can be categorized in many different ways. Based on the winner determination, there are first price and second price actions (Laffont et al. 2001). Based on the manner of operation, there are open and sealed auctions (Laffont et al. 2001). Based on the bidding language, there are combinatorial and simple auctions (Nisan 2000, Yin & Wang 2006). All of these attributes have an influence on the incentive compatibility property of the system as well as the pricing scheme. For simplicity, we start first with the reverse auction, simple bidding language to analyze the effect of the monopoly situation. First price auction is not analyzed because it is not supported by VCG mechanism.

4.1.1 Model Design

The reverse auction consist of one buyer B with one request of A_B amount of resource with the total cost T_B (note that T_B is negative because buyer receives the resource). We define S as the set of sellers, each seller S_i ($0 \leq i \leq n$, where n is the number of sellers) has to offer A_i amount of one resource type with the cost C_i per unit of resource. The use of cost per unit of the seller offer indicates the complementary nature of the resource while the total cost represents the buyer request must be fulfilled as a whole. We sort the set S by C_i in ascending order without lost of generality (i.e. $C_i \leq C_j \forall i < j$).

4.1.2 Winner Determination and Payment Model

The winner determination takes these inputs and compute a set of winners $W = (S_1, S_2, .., Sm) \subset S, m \leq n$. Only the cheapest offers are selected according to the second price auction rule. Each seller will sell part of its resource, i.e. with $a_i \leq A_i$ items. Note that $a_i = A_i \forall i < m$ because of the minimization of the winner determination function described below. We define $V_i = a_i \times C_i$ as the value of the transaction to the seller S_i.

The winner determination function $o(S)$ must satisfy two conditions:

$$o(S) = W : \begin{cases} 1. & min\left(\sum_{i \in W} V_i\right) \\ 2. & \sum_{i \in W} a_i = A_B \end{cases} \quad (1)$$

With this model, the VCG-based payment p_i for seller S_i will be:

$$p_i = \left(\sum_{j \in o(S_{-i})} V_j + T_B\right) - \left(\sum_{j \in W} V_j - V_i + T_B\right) \quad (2)$$

The first element of this equation describes the sum of transaction values of all second winners. The second element describes the sum of transaction values of all first winners excluding S_i. The allocation is possible if the total payments for sellers are smaller or equal to the requested price, or the total transaction value must be at most zero:

$$\sum_{i \in W} p_i + T_B \leq 0 \quad (3)$$

In the case of a budget balanced system, the payment for buyer p_B will be equal to the sum of all seller payments: $\sum_{i \in W} p_i + p_B = 0$.

4.1.3 Monopoly Occurrence Condition

The monopolist seller S_{mono} is included in the set of winners ($S_{mono} \in W$). There are two main cases which lead to monopoly situation:

(i) Without the monopolist, there is insufficient resource to satisfy the request, i.e. there is no possible allocation:

$$\sum_{i \in o(S_{-mono})} A_i < A_B \quad (4)$$

(ii) Without the monopolist, the cost to fulfilled the request is unaffordable to the buyer:

$$\sum_{i \in o(S_{-mono})} V_i + T_B > 0 \quad (5)$$

For both cases, the condition of monopoly occurrence is:

$$\sum_{i \in o(S_{-mono})} V_i + T_B \geq 0 \quad (6)$$

Proof. **Effect of monopoly situation on the allocation**

We examine the payment function for the monopolist. From equation (2), $p_{mono} = (\sum_{j \in o(S_{-mono})} V_j + T_B) - (\sum_{j \in W} V_j - V_{mono} + T_B)$. Due to equation (6),

we have:

$$\Rightarrow p_{mono} \geq -(\sum_{j \in W} V_j - V_{mono} + T_B) \quad (7)$$

To prove the unaffordable allocation, we start by reversing equation (3):

$\sum_{i \in W} p_i + T_B > 0$

$\Leftrightarrow \sum_{i \in W_{-mono}} p_i + p_{mono} + T_B > 0$

Combine with equation (7):

$$\begin{aligned}
&\textstyle\sum_{i \in W_{-mono}} p_i - (\sum_{j \in W} V_j - V_{mono}) \geq 0 \\
\Leftrightarrow\ & \textstyle\sum_{i \in W_{-mono}} (\sum_{j \in o(S_{-i})} V_j) - \\
& \textstyle\sum_{i \in W_{-mono}} (\sum_{j \in W} V_j - V_i) - \\
& \textstyle(\sum_{j \in W} V_j - V_{mono}) \geq 0 \\
\Leftrightarrow\ & \textstyle\sum_{i \in W_{-mono}} (\sum_{j \in o(S_{-i})} V_j) - \\
& (m-1)(\textstyle\sum_{j \in W} V_j) \geq 0 \\
\Leftrightarrow\ & (m-1)(trans.\ values\ of\ 2^{nd}\ winners) - \\
& (m-1)(trans.\ values\ of\ 1^{st}\ winners) \geq 0
\end{aligned}$$

The transaction value of the second winner is always greater than or equal to those of the first winner because the winner determination function $o(S) = W$ picks the minimum transaction values for all possible W on equation (1). Thus we conclude that in any monopoly situation, the allocation is unaffordable. □

4.2 Combinatorial Auction Model

In this section, we define the model of a complex trading system, the combinatorial auction. A simple reverse auction can be viewed as a special case of combinatorial auction, where only one type of resource is traded. We will show how the proof still hold true when shifting from one to multiple resource types.

We gradually map each attribute of the combinatorial auction to a corresponding one in reverse auction, in the same order we defined the reverse auction model.

The first difference is the modelling of seller offers and buyer requests. For each seller S_i, the offer consists of multiple resources and cost tuples S_i^r, where $r \in R$ and R is a set of resource types. Each tuple S_i^r has the amount of offered resource type r, A_i^r and the cost per unit C_i^r. The seller's offer can be described as follow:

$$\begin{aligned}
S_i\ offer &= (S_i^{r1}, S_i^{r2}, .., S_i^{rR}) \\
&= (< A_i^{r1}; C_i^{r1} >, < A_i^{r2}; C_i^{r2} >, \ldots, \\
&\quad < A_i^{rR}; C_i^{rR} >) \quad (8)
\end{aligned}$$

Note that if seller S_i does not have resource type r, then $A_i^r = 0$.

For buyer B, we still have T_B as the total cost of the request (negative value). The total amount of requested resources A_B becomes:

$$A_B = \sum_{r \in R} A_B^r \quad (9)$$

The winner set W is modelled in the same way. The transaction value of seller S_i is: $V_i = \sum_{r \in R} V_i^r$, for each $V_i^r = a_i^r \times C_i^r$ where a_i^r is the amount of sold resource type r of seller S_i.

In the new model of V_i, the winner determination and the payment model are similar to the reverse auction model (1) and (2), without considering the resource types. It is important to eliminate the effects of resource types upon the winner determination and payment scheme to make possible the reuse of the proof for the monopoly situation in section 4.1. Since we still keep T_B, the affordable equation (3) is similar, thus the monopoly situation condition is the same as in (6). We change the monopoly occurrence (4), to reflect resource types:

$$\exists r \in R : \sum_{i \in S\_mono} A_i^r < A_B^r \quad (10)$$

Modelling multiple resource types changes the occurrence rate of the monopoly situation, but since all dependent equations of the proof are unchanged ((1), (2), (3), (6)), we can use it to prove the unallocability of combinatorial auction in a monopoly situation.

4.3 Simple Exchange Model

In this section, we extend the reverse auction model in terms of the number of buyers to make a model for simple exchange. An exchange consists of multiple buyers and sellers in trading for multiple types of resources. A market maker matches buyers and sellers according to the winner determination condition. We define the simple exchange as an exchange of only one type of resource. We make a different set of annotation for simple exchange model, and map this set with corresponding annotations in auction model to reuse the proof. This new set of annotation is also used to construct the proof for simple exchange model.

In the exchange, the set B defines a set of buyers $B_i \in B$. The existence of multiple buyers makes it difficult to use a single variable to represent all the buyers. We used the same identifier space for buyer set B and seller set S. The previous variable x_B is presented as x_i, where $B_i \in B$.

Each buyer request consists of an amount A_i and total cost T_i, where $i \in B$ (note that A_i is now **negative** for buyer). For sellers, an offer has an amount A_i and cost per unit C_i. We define transaction value V_i of S_i or B_i as follows:

$$\begin{aligned}
&\forall i \in B, V_i = T_i \\
&\forall i \in S, V_i = a_i \times C_i
\end{aligned} \quad (11)$$

In the above equations, a_i is the amount of seller S_i that has been sold after the transaction. For buyer B_i, $a_i = A_i$ due to the completeness requirement of the request. The new winner determination conditions are as follows:

$$o(S \cap B) = W : \begin{cases} 1. & min(\sum_{i \in S \cap B}) V_i \\ 2. & \sum_{i \in S \cap B} a_i = 0 \end{cases} \quad (12)$$

These conditions are similar with the auction winner determination condition (1), i.e. the second transaction values are always greater or equal to the first transaction values. There is, however, one difference. In the auction model, there is only one buyer. Therefore, after processing that buyer, the algorithm stops. In an exchange, there must be a stopping condition to halt the algorithm. We sort the buyers using a criteria defined by the system (e.g. FCFS) and iteratively determine the sellers that can supply the request (by taking the offer with smallest costs due to the first condition of (12)). We stop in any of the

following cases. First, when there are not enough resource amounts for the current request. Second, when the smallest cost of the remaining seller exceeds the cost per unit (T_i/A_i) of the current request.

The payment function is similar to the auction payment (2). The difference is only in terms of annotation:

$$p_i = \sum_{j \in o((S \cap B)_{-i})} V_j - (\sum_{j \in W} V_j + V_i) \quad (13)$$

The affordable allocation condition is also different from (3), only in annotation:

$$\sum_{i \in S \cap B} p_i \leq 0 \quad (14)$$

The monopoly occurrence condition is more complicated in exchange model, although the definition is still the same: without the monopolist, the allocation cannot be made. We define two such cases:

(i) There is not enough resources to satisfy the *first request*:

$$A_{B1} + \sum_{i \in o(S_{-mono} \cap B)} A_i < 0 \quad (15)$$

(ii) The resource cost of the winning sellers exceeds the amount affordable by the **first buyer**:

$$U_{B1} + \sum_{i \in o(S_{-mono} \cap B)} V_i > 0 \quad (16)$$

Although the occurrence conditions changed, they yield the same inequality as (6), different only in annotation:

$$\sum_{i \in o(S_{-mono} \cap B)} V_i > 0 \quad (17)$$

As we reconstruct the exchange model from the auction model, we observe that there are only changes in the annotation. Thus we can use the same proof as in section 4.1, removing T_B since it is included in $\sum p_i$, while everything else remains the same.

4.4 An Approach to Allocation under Monopoly Situation

Since VCG is used to achieve strategy proof for a mechanism, we modify the payment scheme to make allocation possible in the monopoly situation. Our aim is to achieve budget balanced in this situation, which VCG is not capable of, while trying to maintain partial strategy proof. Based on the Impossibility Theorem (Myerson & Satterthwaite 1983), it is not possible to achieve at the same time budget balanced, strategy proof and allocation efficient properties in a payment scheme. Although a system with partial strategy proof does not inhibit participants to lie about their true preferences, it will make it difficult for them to gain any benefit. Furthermore, the strategy proof is not necessary in the monopoly situation when there is a lack of competitive.

We propose new payment function for allocation under the monopoly situation. It can be applied to a wide range of trading systems including auction, combinatorial auction and exchange. All such systems can include this new function to address the monopoly situation while keeping its own payment system on the other cases. By doing that, the system improve its performance without lost of economic efficiency.

An obvious way to make an allocation is to pay each participant exactly their requested value. But this method is not even partially strategy proof since everyone knows the payment in advance and thus lead to the winner's curse, like in first price auction. Furthermore, this method is not budget balanced.

Our propose payment function in the monopoly situation designed to achieve budget balanced and allocation efficiency with partial strategy proof:

$$\begin{aligned} \forall i \in S : p_i &= f \times V_i \\ \forall i \in B : p_i &= \tfrac{f}{E} \times V_i \end{aligned} \quad (18)$$

E is the ratio of the total buyer preference and total seller preference:

$$E = \frac{-\sum_{i \in B} V_i}{\sum_{i \in S} V_i} \quad (19)$$

The partial strategy proof is achieved through the function f. Any function that satisfied the constraint can be used:

$$1 \leq f \leq E \quad (20)$$

There are many such functions which fit into these constraints, e.g. $f = (E+1)/2$, $f = \sqrt{E}$, $f = b^{\log_a E}$ $(0 \leq b \leq a \leq 1))$, etc. The use of linear function like $f = (E+1)/2$ leads to predictable payments, given enough information. In this case, the payment for each buyer/seller is proportional with its preference, and the total payment of buyers/sellers is the middle value of the total preferences of buyers and sellers. However, this is not possible with non-linear functions, and thus makes it hard for any participant to manipulate the final payment. In the case of sealed bids trading systems, a popular implementation, such manipulation is impossible for any function f used.

Proof

The payment scheme is budget balanced and affordable by the buyers if it satisfy the constraint:

$\sum_{i \in S} V_i \leq \sum_{i \in S} p_i = -\sum_{i \in B} p_i \leq -\sum_{i \in B} V_i$

The results can be deduced directly:

$1 \leq f \Rightarrow \sum_{i \in S} V_i \leq \sum_{i \in S} p_i$

$f \leq E \Rightarrow -\sum_{i \in B} p_i \leq -\sum_{i \in B} V_i$

Equation (19) $\Rightarrow \sum_{i \in S} p_i = -\sum_{i \in B} p_i$

Example: In the following example we use $f = (E+1)/2$.

Buyers		**Sellers**	
A_i	V_i	A_i	V_i
4	-5	7	3
4	-4	1	2

Table 1: An example of a monopoly situation

Seller 1 is the monopolist, since it hold 7 out of 8 resource units. Without the monopolist, there is no possible allocation. Using our scheme, the payments are:

Buyers	**Sellers**
-3.9	4.2
-3.1	2.8

Table 2: Payments in monopoly situation

These payments are budget balanced ($-3.9-3.1+4.2+2.8=0$) and affordable ($-3.9>-5$, $-3.1>-4$, $4.2>3$, $2.8>2$). We also observe that $\sum_{i\in S} p_i = \frac{1}{2}(\sum_{i\in S} V_i - \sum_{i\in B} V_i)$, i.e., payments are divided equally between sellers and buyers. We can use any function in family f such as:

$$f = 1 + (E-1) \times e, 0 \leq e \leq 1 \qquad (21)$$

If $e = \frac{1}{2}$, the distribution of payment is equal between sellers and buyers. If $e < \frac{1}{2}$, the total payments of buyers is higher. If $e > \frac{1}{2}$, the total payments of sellers is higher. A payment system can vary e to motivate sellers or buyers based on the supply and demand. For example, if the monopoly happened at over-demand condition, the system can adjust e to motivate more sellers to participate in the market and alleviate the monopoly situation.

5 Simulation Results and Analysis

In the previous section, we have shown the effect of monopoly for the winner determination process in Vickrey-based mechanism and proposed a new payment method. In this section, we quantify the occurrence rate of this effect through simulation.

We choose combinatorial exchange as the model to measure the occurrence rate because the result obtained is also representative in auction and combinatorial auction. We ran a number of simulations with varying market size. By changing the ratio of demand (buyers) and supply (sellers) we model different market conditions such as under-demand, balanced market and over-demand.

We use jCase (Schnizler & Neumann 2007) simulator where different winner determination and payment computation algorithms for the combinatorial exchange model can be simulated. For simplicity of analysis, we assume one resource type only and use the uniform distribution for generating the number of buyers and sellers, number of units per resource requested by buyers and offered by sellers, cost per unit of resource, etc. The simulation were performed on a cluster of ten nodes, each with two quad-cores 1.83GHz Xeon processors and 4GB of RAM.

Using the Vickrey payment function, unless the total value of the transactions is affordable, as defined in (3), we count the allocation as a monopoly situation. The percentage of monopoly occurrences (occurrences %)) is the ratio of occurrences over total number of resource allocations.

As shown in Figure 1, the occurrence rate is high when the market size is small. It approaches zero as the market grows because the probability of sellers becoming monopolistic diminishes. In Figure 2, we model various market conditions by varying the ratio of buyers and sellers. When resource demand is high, sellers create a monopolistic market. In summary, the monopoly effect is higher when the size of the market is small. This has a bearing for Internet-based system such as a Grid when the market is dynamic and may be in constant state of flux. We have shown that pricing is possible and allocation can be made under such situation.

6 Conclusion

In this paper, we exposed a deficiency of the well-known VCG mechanism that have not been studied. When sellers create a monopolistic situation, the second price cannot be determined and resource allocation cannot be performed. We showed that this

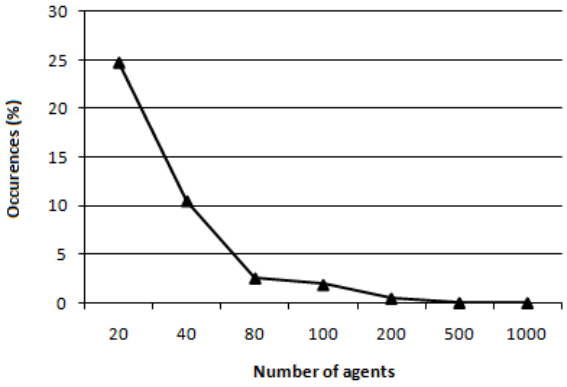

Figure 1: Percentage of Monopoly Occurrences with Varying Market Size

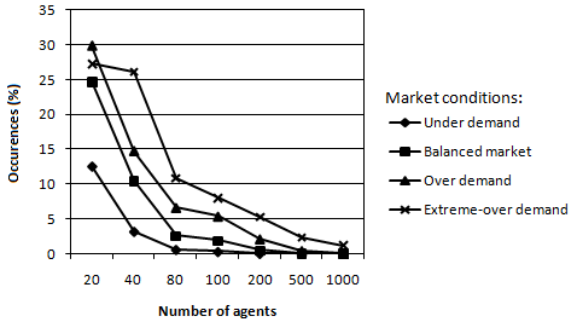

Figure 2: Percentage of Monopoly Occurrences under Different Market Conditions

situation arises with high occurrence rate in many market-based algorithms that adopt the VCG pricing scheme. We propose a new pricing method for making resource allocation under the monopoly situation that achieve budget balanced and economic efficiency. Because of the lack of competition in such situation, our method relaxes the strategy proof property. The proposed method also has the advantage of steering the market back to VCG allocation. When market condition remain under-demand, monopolistic situation can persist. We are investigating the incentive compatible property under repeated monopolistic allocations and are working towards a strategy proof.

Acknowledgements
This work is supported by the National University of Singapore under grant number R-252-000-339-112.

References

Braynov, S. & Sandholm, T. (2003), 'Auctions with untrustworthy bidders', *Proc. of International Conference on E-Commerce* **0**, 363.

Bubendorfer, K. (2006), Fine grained resource reservation in open grid economies, *in* 'Proc. of the 2nd IEEE International Conference on e-Science and Grid Computing', IEEE Computer Society, Washington, DC, USA, p. 81.

Chun, B. N. & Culler, D. E. (2000), Market-based proportional resource sharing for clusters, Technical report, Berkeley, CA, USA.

Dash, R. K., Jennings, N. R. & Parkes, D. C. (2003), 'Computational-mechanism design: A call to arms', *IEEE Intelligent Systems* **18**(6), 40–47.

Dramitinos, M., Stamoulis, G. D. & Courcoubetis, C. (2005), 'Auction-based resource allocation in

umts high speed downlink packet access', *Next Generation Internet Networks* pp. 434–441.

Green, J. & Laffont, J. J. (1977), 'Characterization of satisfactory mechanisms for the revelation of preferences for public goods', *Econometrica* **45**(2), 427–38.

Kothari, A., Sandholm, T. & Suri, S. (2004), 'Solving combinatorial exchanges: optimality via a few partial bids', *Proc. of the 3rd Joint Conference on Autonomous Agents and Multiagent Systems* **3**, 1418–1419.

Laffont, R., Stockinger, H., Giddy, J. & Abramson, D. (2001), Economic models for management of resources in peer-to-peer and grid computing, Computational Economics 0108001, EconWPA.

Lavi, R. & Swamy, C. (2005), Truthful and near-optimal mechanism design via linear programming, *in* 'Proc. of the 46th Annual IEEE Symposium on Foundations of Computer Science', IEEE Computer Society, Washington, DC, USA, pp. 595–604.

Myerson, R. B. & Satterthwaite, M. A. (1983), 'Efficient mechanisms for bilateral trading', *Journal of Economic Theory* **29**(2), 265–281.

Neumann, D., Schnizler, B., Weber, I. & Weinhardt, C. (2007), Second-best combinatorial auctions - the case of the pricing-per-column mechanism, *in* 'Proc. of the 40th Annual Hawaii International Conference on System Sciences', IEEE Computer Society, Washington, DC, USA, p. 33.

Nisan, N. (2000), Bidding and allocation in combinatorial auctions, *in* 'Proc. of the 2nd ACM Conference on Electronic Commerce', ACM, New York, NY, USA, pp. 1–12.

Nisan, N. & Ronen, A. (2001), Algorithmic mechanism design, *in* 'Games and Economic Behavior', Vol. 35, Academic Press, pp. 166–196.

Parkes, D. C., Cavallo, R., Elprin, N., Juda, A., Lahaie, S., Lubin, B., Michael, L., Shneidman, J. & Sultan, H. (2005), Ice: an iterative combinatorial exchange, *in* 'Proc. of the 6th ACM Conference on Electronic Commerce', ACM, New York, NY, USA, pp. 249–258.

Parkes, D. C., Kalagnanam, J. R. & Eso, M. (2001), Achieving budget-balance with Vickrey-based paymentschemes in exchanges, *in* 'Proc. 17th International Joint Conference on Artificial Intelligence', pp. 1161–1168.

Rogers, A., Dash, R. K., Jennings, N. R., Reece, S. & Roberts, S. (2006), Computational mechanism design for information fusion within sensor networks, *in* 'Proc. of 9th International Conference on Information Fusion', Italy.

Schnizler, B. & Neumann, D. (2007), 'Combinatorial exchanges for coordinating grid services', *SIGecom Exch.* **7**(1), 65–68.

Takahashi, E. & Tanaka, Y. (2003), 'Auction-based effective bandwidth allocation mechanism', *Proc. of the IEEE Inter. Conf. on Telecommunications* **2**, 1046–50.

Yin, H. & Wang, X. J. (2006), 'Optimal mechanism design for multi-objective double auction', *Proc. of the Fifth Inter. Conf. on Machine Learning and Cybernetics* .