To Play or to Control: a Game-based Control-theoretic Approach to Peer-to-Peer Incentive Engineering

Weihong Wang, Baochun Li

Department of Electrical and Computer Engineering
University of Toronto
{*wwang, bli*}*@eecg.toronto.edu*

Abstract. In peer-to-peer applications, we need to encourage selfish users to share and contribute local resources to the global resource pool that all peers may benefit from, by providing adequate *incentives*. If we assume that all users are non-cooperative and always attempt to maximize their own *net gains*, at the first glance, we could model such behavior as a non-cooperative game and derive the equilibrium that no users deviate from. However, two observations complicate the case. (1) In such a game, user valuation on the contribution amount fluctuates, due to the dynamic supply-demand relationship of the shared resources; and (2) desirable global system properties require payoff functions to be reasonably designed. In this paper, we model the peer-to-peer system as a *Cournot Oligopoly* game with dynamic payoff functions that incorporate system performance requirements, and propose a control-theoretic solution to the problem. Throughout the paper, we use a peer-to-peer global storage system as a running example and case study. Simulation results have shown that the control-theoretic solution may effectively adapt the user contributions to track system dynamics, maximize the local net gain, and achieve satisfactory global properties.

1 Introduction

In peer-to-peer networks, each peer host contributes its local resources to serve the common good, and may benefit from resources contributed by other peers in return. In peer-to-peer storage systems (*e.g.*, CFS [1], OceanStore [2], PAST [3]), peers contribute their local storage space and network bandwidth to the system, and are granted rights to store (or backup) data in the global storage pool. Similarly, other peer-to-peer applications may require peers to contribute network bandwidth (e.g., Resilient Overlay Networks [4]) or CPU cycles (e.g., the concept of Grid computing [5]). Based on such a fundamental design philosophy, peer-to-peer applications provide appealing features of enhanced system robustness, high service availability and scalability.

However, all is not rosy. The critical observation that users are generally *selfish* and *non-cooperative* may severely undermine the expected peer-to-peer structure. For example, the *free rider* phenomenon has been revealed [6, 7] in peer-to-peer file sharing applications such as Gnutella: most users are selfish and never share any local files, such that the peer-to-peer system is only supported by a small group of *supernodes*, and degrades to a client-server-like centralized structure. The root cause of the problem is, obviously, there exist no *incentives* for users to be altruistic. Therefore, if we assume that all users are selfish and wish to maximize their own net gains at all times, engineering incentives is a must to encourage contribution and maintain the robustness and availability of peer-to-peer systems.

The question, now, turns to how incentives may be designed. We may naturally be led to game theory for two reasons. First, incentives and costs are natural components of the users' *net gains*, which may be easily modeled by *payoff functions* in game theory. Second, the selfishness of users guarantees that they seek to maximize their gains, which conforms with the fundamental assumptions in game theory as well. Game theory studies whether an equilibrium exists in a game, and if so, how to derive such an equilibrium. However, the question at hand leads to the issue of how we may *construct* (or *design*) the payoff functions in a game, such that *certain desired global properties* may be achieved once the users reach their respective equilibria. This is in the domain of *inverse game* (or *mechanism design* [8]), which is usually hard to solve.

Even if the payoff functions are designed, we need an adequate solution to drive the users toward the desirable equilibrium. In this case, system dynamics have to be incorporated into the incentives and costs that constitute such *time-varying* payoff functions. On one hand, user contributions dynamically affect the global system states (*e.g.*, total amount of resources contributed); on the other hand, to maintain acceptable system performance, their decisions on the amounts of contributed resources must be adjusted over time based on the observed and predicted system dynamics. In order to assist users to make such time-varying decisions with the presence of system uncertainties, a control-theoretic approach seems more adequate than game theory.

Towards peer-to-peer incentive engineering, should the users play games with specifically constructed payoff functions, or should they be controlled to make time-varying decisions? In this paper, we attempt to combine the benefits of both worlds. We design the payoff function in a game-theoretic perspective, such that it explicitly incorporates the desirable global system properties. We then use such designed payoff function as the *objective function* in an optimal control system deployed at every peer. The optimal control system makes decisions on the quantity of contribution to the global pool, such that the objective function is maximized, subject to certain constraints. Our game-based controller design may effectively adapt the user contributions to track system dynamics, maximize the local net gain, and achieve satisfactory global system performance. In simulation studies, we compare the optimal control based solution with the pure game-based solution. Throughout the paper, we use a peer-to-peer global storage system as a running example and case study.

The remainder of the paper is organized as follows. Sec. 2 presents preliminaries regarding our system models and objectives. A game-theoretic perspective of the system is illustrated in Sec. 3, which evolves into a control-theoretic approach presented in Sec. 4. The control performance of the proposed mechanisms is evaluated in Sec. 5. Related work is discussed in Sec. 6, and Sec. 7 concludes the paper.

2 Preliminaries

In this paper, we are concerned about engineering sufficient *incentives* for users in peer-to-peer systems, so that contribution of local resources to the common pool is encouraged. We assume that all users are selfish, in the sense that they seek to maximize their net gains at all times once the incentives are provided. We believe that the *net gain* of a user is equivalent to the offered incentives subtracted by the *cost* in providing the resources.

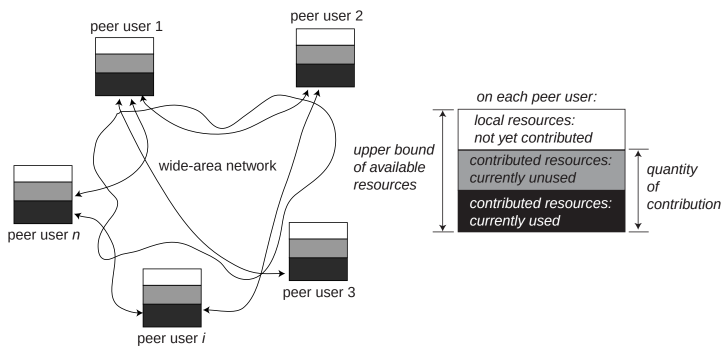

Fig. 1. Model of a peer-to-peer system: contributions, usage and bounds

Fig. 1 shows a detailed view of available resources on each of the peer users. The type of resource may be bandwidth, storage space or CPU cycles, depending on the features of the particular peer-to-peer application. On each peer user, there exists an *upper bound of available resources*, *e.g.*, the maximum available storage space. The user makes one simple decision: what is the quantity of resource it should contribute to the globally shared pool? The contributed resource may not be utilized at all times, and may be further divided into used and unused portions.

Without loss of generality, we concentrate on the case study of a peer-to-peer storage system throughout the paper, where the local *storage space* is the resource to be contributed to the common pool. We proceed to present a brief description of such an application.

Peer-to-peer storage systems are designed to aggregate contributed storage space distributed over a network, in order to conserve the cost of proprietary high-capacity and centralized storage devices. A good motivating example of using such a system is off-site file system backups. A peer-to-peer storage system is by no means static: peer users join and leave the network freely, and may dynamically adjust the quantity of contributions. The size of the global pool of storage space varies over time, and such dynamic behavior has become the subject of extensive research (e.g., CFS [1]). In some of the existing proposals, user data files are stored in the granularity of *blocks*, which are scattered into the contributed storage of multiple peers. Such a strategy is beneficial to achieve load balancing, hotspot relief and robustness of data.

For our case study of peer-to-peer storage systems, we make the following reasonable assumptions on an abstract level:

(1) *Adaptive quantity of contribution.* Peer users are allowed to adjust the quantities of their contributed resources (*i.e.*, storage space), in order to maximize their own net gains at all times. In this case, the contributed resources may be reclaimed at a later time. It is realistic to make such an assumption, since peer users are allowed to join or depart the system at will.

(2) *Hash-based location and lookup of data blocks.* We believe that the most effective peer-to-peer storage system utilizes *distributed hash tables* (DHTs) to hash data blocks into identifiers, and these identifiers are used to determine which peer users are going to store these blocks[1]. In this case, operations on data blocks in one peer user are obviously independent from other geographically nearby users.

(3) *Proportional usage of contributed capacity.* Effective peer-to-peer storage systems employ mechanisms to achieve almost ideal load balancing with respect to contributed but unused storage resources. For example, CFS employs *virtual servers* with approximately the same storage capacity, to guarantee that the usage of the contributed storage of every user is approximately in proportion to its amount of contribution.

3 Incentives: a Game-Theoretic Perspective

Since each selfish user seeks to maximize its own net gain, *i.e.*, the incentives minus the cost of contributions, it is natural to model the system from a game-theoretic perspective. Game theory addresses multi-person decision making problems, in which the *players* are autonomous and rational decision makers. Each player has a set of *strategies* at its disposal, and a *payoff function* (its valuation of each combination of strategies of all the players). Players choose strategies to maximize their own payoffs, with the consideration that their payoffs are affected by the decisions of all the players. Thus, information about others' strategy and payoff functions is critical in any game.

A *Nash equilibrium* identifies a stable state of the game, at which no player can improve its payoff by deviating from the state, if no other players do so. At the equilibrium, each player receives a payoff that is optimal with respect to the player's knowledge or belief about all other players' strategies. Formally, the condition of Nash equilibrium may be expressed as follows.

$$u_i(s^*_{-i}, s^*_i) \geq u_i(s^*_{-i}, s_i),\ \forall s_i \in S_i,\ \forall i \in N \tag{1}$$

where S_i is player i's strategy set, s^*_i is the strategy selected by player i at equilibrium, s^*_{-i} is the strategies selected by all the other players at equilibrium, and u_i is the payoff function for player i. Further, a *static game* characterizes the situation in which all players make decisions simultaneously without following any particular sequence of play, and decisions are made once for all. A *repeated static game* extends a static game in a stage by stage manner.

3.1 The Cournot Oligopoly Game

Assume that time is discretized, we model the decision-making procedure in the peer-to-peer system as a repeated static game, with each stage of the game corresponding to a time slot. In our case, the players correspond to the peer users, and the strategy space of each peer user i is represented by $S_i(k) = [0, C_i(k)]$, where $C_i(k)$ denotes the upper bound of available resources for user i during the time slot k. When a peer user makes a decision on the quantity of contributed resource, it has selected a strategy within the strategy space $S_i(k)$.

[1] The reader is referred to the design of CFS [1] for further details.

Within each stage, the game closely resembles the *Cournot Oligopoly* game. In a Cournot Oligopoly game, n firms act as players, each rationally decide their own productions $\{q_i\}$ of a homogeneous product in the market. Let $Q = q_1+\cdots+q_n$ denote the aggregate quantity on the market, and a denotes the total demand. The market-clearing price $P(Q)$ is given by the inverse demand relationship: $P(Q) = a - Q$ (assuming $Q < a$, otherwise $P = 0$). Assume all firms have the same marginal cost c (assume $c < a$), and no fixed costs exist.

Before exploring the Nash equilibrium of the *Cournot Oligopoly* game, we resort to the simpler case of a *Cournot Duopoly* game, where only *two* firms are present. In this case, the profit of either firm can be expressed by the following payoff function ($i, j = 1, 2$):

$$u_i(q_i, q_j) = q_i \cdot P(q_i + q_j) - q_i \cdot c = q_i[a - (q_i + q_j) - c] \quad (2)$$

The game assumes that both firms know that their rival has the same knowledge of the market-clearing price $P(Q)$ and produces at the same cost (a static game with complete information in game theory), hence, either firm is able to decide their optimal quantities of production by solving the following equation array:

$$\begin{cases} \frac{\mathrm{d}u_1}{\mathrm{d}q_1} = a - 2q_1 - q_2 - c = 0 \\ \frac{\mathrm{d}u_2}{\mathrm{d}q_2} = a - 2q_2 - q_1 - c = 0 \end{cases} \quad (3)$$

And the derived Nash equilibrium is symmetric: $q_i = (a - c)/3$.

Fig. 2 (A) illustrates such an equilibrium. Similarly, it can be shown that the Nash equilibrium of the *Cournot Oligopoly* game is $q_i = (a - c)/(n + 1)$, where n is the number of firms as players. It is easy to see that, if n is sufficiently large, and c is relatively small compared with a, the total quantity of production on the market approaches a.

3.2 Designing the Payoff Function

In peer-to-peer systems under consideration, we aim to encourage appropriate user contributions, rather than *excessive* contributions. In this case, we need to reach a balanced trade-off between two objectives: to provision adequate total resources in order to accommodate unpredictable future service requirements with high probability, and to maintain high resource utilization levels. We therefore need to properly design the payoff function, so that the *incentive* a user receives for its quantity of contribution, as well as the associated *costs*, reflect the actual system dynamics.

In this paper, we model peer-to-peer resource contribution systems as a variant to the Cournot Oligopoly game due to the following reasons. First, in such peer-to-peer systems, each peer user i decides its quantity of contribution within the strategy space of $S_i(k) = [0, C_i(k)]$, with an incentive that is dependent on the contribution of all users. The semantics is identical to that of the Cournot Oligopoly game. Second, the market-clearing price in the Cournot Oligopoly game is based on the principle of reverse demand, and naturally regulates user behavior based on the supply-demand relationship.

Despite such similarities, we still need to tailor the definitions of a and c to the specific requirements of the context. The term a, in the expression of market-clearing price in the Cournot Oligopoly game, has been used to represent the maximum demand on the market, and thus regulates the maximum achievable total production in the game. In peer-to-peer systems, we let a reflect the *total desirable quantity of resource*. Then the market-clearing price is expressed as the difference between a and the *total contribution* of resources, according to the inverse demand principle as in the Cournot Oligopoly example. The *incentives* towards user contribution is thus the market-clearing price multiplied by the quantity of resource contribution.

It should be noted that it is hard to determine the total desirable quantity of resources off-line, since peers may join and leave the system at will, and usage in the shared resource pool is unpredictable. However, the quantity can be estimated based on the observed history, if it varies relatively slowly over time, which is the strategy we utilize.

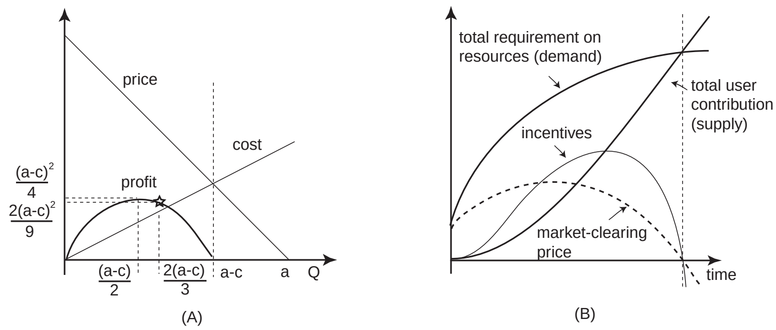

Fig. 2. (A) The Cournot Duopoly game. The star denotes the Nash equilibrium. $\frac{a-c}{2}$ is the optimal output quantity in the monopoly case. (B) The relationship among the total desirable quantity of resource (or demand), total resource contribution (or supply), the market-clearing prices and the engineered incentives (market-clearing price multiplied by the quantity of user contributions).

Fig. 2(B) shows that, when the market-clearing price and the user's quantity of contribution change over time, there exists a point, after which the incentives offered to the peer user start to decrease. This implies that when there are excessive resources available in the global pool, peers should no longer benefit further by contributing additional resources.

Beyond incentives, contributing valuable local resources to the global pool comes with costs. In designing the marginal cost c in the payoff function, a heuristic choice is to assume a constant cost per unit quantity, as in the original Cournot Oligopoly game. However, in realistic peer-to-peer resource sharing applications, such an assumption may not reflect the true associated costs. For example, in the case study of peer-to-peer storage systems, the overwhelming cost of sharing each unit of storage space may not be the local storage *per se*, the peer users may be much more concerned with the local *bandwidth* being consumed by other peers *accessing* data contained in the contributed storage space.

From a user's perspective, taking the quantity of contribution as the *cause*, the *effects* of the contribution may be reflected by the observable bandwidth consumption, as is shown by Fig. 3. Though both the cause and effects are locally observable, the system dynamics governing these parameters are, unfortunately, not known.

We now proceed to formally design the payoff function, which engineers appropriate incentives and models the true costs. With the peer-to-peer storage system as our

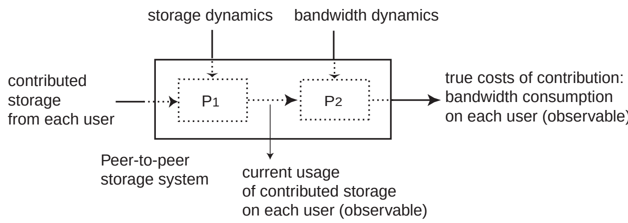

Fig. 3. The cause-effect relationship between the quantity of contribution, the contributed storage being used, and the bandwidth consumption due to such usage.

example, we list all the important notations to be used in this paper in Table 1. Since we assume discrete time domain, the variables $s_i(k)$ and $b_i(k)$ in Table 1 represent their respective average values during slot k, and are observed at the end of slot k. The quantity of contribution $c_i(k)$, as well as the upper bounds $C_i(k)$ and $B_i(k)$, are determined at the beginning of slot k, however.

We design the term a as $\lambda \sum_j s_j(k)$, which represents the desirable quantity of total storage (or *system capacity*). λ is a system wide parameter that defines the desired level of global *storage utilization*. Then, market-clearing price is expressed by $\lambda \sum_j s_j(k) - \sum_j c_j(k)$, and the incentives that user i receives for contributing $c_i(k)$ is:

$$c_i(k) \cdot [\lambda \sum_j s_j(k) - \sum_j c_j(k)] \tag{4}$$

Table 1. Case study of peer-to-peer storage systems: notations

Symbols	*Explanations*
N	The total number of peer users
$c_i(k)$	The storage contribution of user i in slot k
$C_i(k)$	The upper bound of $c_i(k)$
$s_i(k)$	The amount of occupied storage contribution of user i in slot k
$b_i(k)$	The bandwidth consumption on user i in time slot k
$B_i(k)$	The upper bound of acceptable $b_i(k)$
$c^{-i}(k)$	$\sum_{\substack{j\in N\\ j\neq i}} c_j(k)$
$s^{-i}(k)$	$\sum_{\substack{j\in N\\ j\neq i}} s_j(k)$

Since increased storage contribution results in increased bandwidth consumption by other peers, which is highly undesirable for the user, we model the user's reluctance towards further contributions as an exponentially increasing function of its bandwidth consumption: $[b_i(k)/B_i(k)] \cdot e^{b_i(k)/B_i(k)}$. Thus, the higher the relative bandwidth consumption $b_i(k)/B_i(k)$, the higher the marginal cost.

We now have finalized the payoff function, denoted by $u_i(k)$ for user i and in time slot k, as follows:

$$u_i(k) = c_i(k) \cdot \left\{ \lambda \sum_j s_j(k) - \sum_j c_j(k) - \frac{b_i(k)}{B_i(k)} \cdot e^{b_i(k)/B_i(k)} \right\}. \quad (5)$$

In the repeated static game we have defined, for each time slot k, every user i attempts to adjust $c_i(k)$, so as to maximize its payoff in the stage, based on the prescribed payoff function in Eq. (5). Comparing with Eq. (2), we may observe that the terms a and c are time-variant quantities, which are dependent on the actual system states. In addition, when the relative bandwidth consumption $b_i(k)/B_i(k)$ is sufficiently low, the marginal cost is negligible compared to the market-clearing price ($c \ll a$). Thus, the total storage contribution approximates $a \cdot N/(N+1) = \lambda \cdot N \sum_j s_j(k)/(N+1)$, which means that the desirable system storage utilization can be adjusted by tuning the value of λ. Assume an average storage utilization of $(66 \sim 80)\%$, the corresponding λ would be $(1.25 \sim 1.5)(N+1)/N$.

As a simple illustration, the functions of the aforementioned *incentives*, *costs* and *payoffs* are shown in Fig. 4.

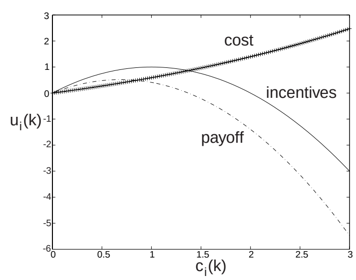

Fig. 4. The payoff function, including the incentives and costs.

4 Game-Based Optimal Control

Our process of engineering the incentives and the costs (that constitute the payoff function) is identical to that of *inverse game theory* (or *mechanism design*), where the payoff function is designed, so that certain desirable global properties are achieved at the Nash equilibrium. It may seem that we may now apply standard game theory and investigate the properties achievable at Nash equilibrium, once the equilibrium point is derived by solving a group of maximization problems.

However, there exist fundamental difficulties along this path. In the game we have designed, the payoff functions of Eq. (5) are not only heterogeneous for different users due to the user-specific parameter $B_i(k)$, but also time-varying: users locally estimate $b_i(k)$ and $s_i(k)$, based on their views of the global storage system updated on-the-fly.

Besides, each user makes its decision on $c_i(k)$ at the beginning of time slot k. However, at this time instant, $s_i(k)$ and $b_i(k)$, which denote their respective average value during slot k, are not yet obtainable. What we are able to obtain, instead, are their old values, $s_i(l)$ and $b_i(l)$, $l \leq k-1$. A direct solution might be to use $s_i(k-1)$ and $b_i(k-1)$ in lieu of $s_i(k)$ and $b_i(k)$, respectively. However, considering the length of time slot k, such an estimation may be taken as a convenience at best.

In addition, from the game-theoretic perspective, the decision making procedure potentially requires each peer to know the parameters (*e.g.*, $s_i(k)$ and $b_i(k)$) of the payoff functions of all other peers before deciding their $c_i(k)$. However, such information cannot be exchanged beforehand, since $s_i(k)$ and $b_i(k)$ are still unavailable at the time of decision making. Therefore, the exact payoff function of any user remains unknown to all other users, which is different from the case of the Cournot Oligopoly example.

Despite these difficulties, what we do need to know is the relationship between (1) the optimal quantity of contribution $c_i(k)$, determined at the beginning of slot k; and (2) $s_i(k)$ and $b_i(k)$ to be observed during the same time slot. Such a relationship, apparently, is determined by the behavior of the external system that we have not investigated so far.

When deciding a new value of $c_i(k)$, it is possible for each user to dynamically identify a mathematical model for the external system based on its locally observed values of $s_i(k)$, $b_i(k)$ and $c_i(k)$. On one hand, given any $c_i(k)$, new values of $s_i(k)$ and $b_i(k)$ can be predicted, so that the objective function can be evaluated, and the optimal value of $c_i(k)$ can be calculated; on the other hand, since new decisions are made on the basis of the model, the user's strategy space is in fact restricted to a set that is closer to the probable system behavior. In this way, we are naturally led to a control-theoretic solution to the game.

Furthermore, due to the difficulty for users to promptly exchange information about their current payoff function, we propose that users determine new quantities of contributions based on other users' status (*i.e.*, $s_i(k)$, $c_i(k)$) in the previous time slot, so that they make decisions according to their observations on, rather than inference about, other users' behavior.

In this section, we propose a control-theoretic approach to address these problems. We design a decentralized optimal control system in the game setting, such that the payoff function Eq. (5), which has incorporated global system performance objectives — in the market-clearing price and marginal cost terms — are taken as the *objective function*. The control law, which is equivalent to the trajectory of contribution decisions, is derived as the maximizing solution to the objective, subject to constraints of the system model. Therefore, we utilize users' selfishness in maximizing their own payoff, and achieve the following goals simultaneously: (1) achieving sufficient total storage capacities; (2) maintaining high storage utilization; and (3) avoiding severe bandwidth stress at participating peers.

4.1 Design of the Optimal Control System

Towards our aforementioned objectives, the peer users rely on decentralized optimal controllers to locally adapt their decisions on their quantities of contribution. Fig. 5 shows the block diagram of the optimal control system design.

For the local control system at peer user i, the entire global peer-to-peer system is the *plant* to be controlled. However, any single peer user i acts only as a *port* to the plant, and $c_i(k)$ is the only control it may impose on the plant. $s_i(k)$ and $b_i(k)$ are affected not only by peer i's contribution quantity $c_i(k)$, but also by the quantities of contribution by other users, as well as unknown system dynamics that are beyond control of our model.

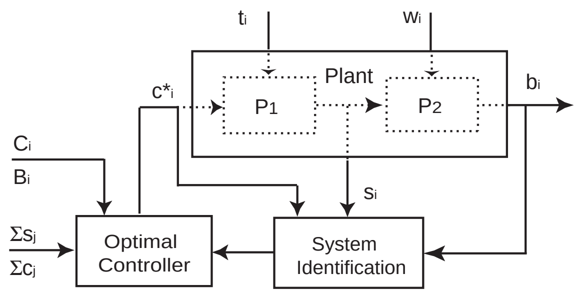

Fig. 5. The block diagram of the decentralized optimal control system.

More concretely, we consider the system seen by user i as a discrete time-varying linear system, with $c_i(k)$, $b_i(k)$, and $s_i(k)$ as the input, output, and state variables, respectively. System dynamics with regard to storage usage and bandwidth consumption, which are caused by the insertion, deletion and retrieval of data blocks, are modeled as random noises.

The problem can be further formulated as a decentralized optimal control task: each user i decides its optimal input trajectory $c_i(k)$ to the plant, which maximizes its playoff function as shown in Eq. (5), subject to the constraints given by Eq. (6):

$$c_i(k) = \arg\max u_i(c_i(k), c^{-i}(k))$$
$$\begin{cases} b_i(k) = F(c_i(k)) \\ c_i(k) \leq C_i(k) \\ b_i(k) \leq B_i(k) \end{cases} \tag{6}$$

where $b_i(k) = F(c_i(k))$ represents the stochastic model of the plant. As it becomes apparent, the correct identification of the plant is critical to the optimal control system.

4.2 The Plant: System Identification

From a control-theoretic perspective, we model the plant as a discrete-time stochastic linear system. A state space model can be formulated as follows:

$$\begin{cases} A(q^{-1})s_i(k) = B(q^{-1})c_i(k) \\ C(q^{-1})b_i(k) = D(q^{-1})s_i(k) \end{cases}$$

where $s_i(k)$ is the state variable, and the form $R(q)$ stands for a polynomial in the *forward shift operator* q, for instance, given $R(q^{-1}) = 1+2q^{-1}+q^{-2}$, $R(q^{-1})s_i(k) = s_i(k) + 2s_i(k-1) + s_i(k-2)$.

Further, we have assumed that at any time slot k, the block insertion rate at user i is roughly in proportion to its unused share of storage contribution (Sec. 2); the bandwidth consumption $b_i(k)$ is essentially dependent on the observed usages at user i (subject to uncertain factors). Therefore, the system model can be refined as:

$$\begin{cases} s_i(k) = s_i(k-1) + \alpha_i[c_i(k) - s_i(k-1)] - \beta_i s_i(k-1) + t_i(k) & (7.1) \\ b_i(k) = \gamma_i(q^{-1})s_i(k) + w_i(k) & (7.2) \end{cases} \tag{7}$$

where $\alpha_i[c_i(k) - s_i(k-1)]$ stands for the amount of inserted data at user i in slot k, $\beta_i s_i(k-1)$ is the amount of deleted data. Apparently, α_i and β_i are time-varying parameters, and the uncertainties in their variations are accounted for by the zero-mean *white noise* term $t_i(k)$. Similarly, the coefficients of $\gamma_i(q^{-1})$ also change over time, and the zero-mean white noise $w_i(k)$ represents the uncertain factors from the global system regarding bandwidth consumption at user i.

We adopt the *stochastic approximation* (SA) algorithm [9] to estimate the unknown parameters α_i, β_i and the coefficients of $\gamma_i(q^{-1})$, based on the observed values of $s_i(k)$, $c_i(k)$ and $b_i(k)$.

Consider Eq. (7.1) as an example. The equation can be written in the form of: $y(k) = \boldsymbol{\phi}^T(k)\boldsymbol{\theta} + t_i(k)$, where $y(k) = s_i(k)$, $\boldsymbol{\phi}^T(k) = (s_i(k-1), c_i(k) - s_i(k-1))$, both consist of observable variables, $\boldsymbol{\theta}$(k) $= (1 - \beta_i(k), \alpha_i(k))$ contains the parameters to be estimated, and $t_i(k)$ is the unknown noise.

The estimated parameter vector $\widehat{\boldsymbol{\theta}}(k)$ is derived as the minimizing solution to the least-squares loss function $V(\boldsymbol{\theta}, t) = \frac{1}{2}\sum_{i=1}^{t}(y(i) - \boldsymbol{\phi}^T(i)\boldsymbol{\theta})^2$. A simplified recursive algorithm is given by

$$\widehat{\boldsymbol{\theta}}(k) = \widehat{\boldsymbol{\theta}}(k-1) + \epsilon P(k)\boldsymbol{\phi}(k)(y(k) - \boldsymbol{\phi}^T(k)\widehat{\boldsymbol{\theta}}(k-1))$$

where

$$P(k) = \left(\sum_{i=1}^{k}\boldsymbol{\phi}^T(i)\boldsymbol{\phi}(i)\right)^{-1}$$

ϵ is the *adaptation gain* which tunes the adjustment step of the estimates.

For the sake of predicting system dynamics, it is advantageous to have the estimated model parameters update slower than system variables. We take the following averaging technique to achieve *smoother* parameter variations: $\widehat{\boldsymbol{\theta}}(k) = \widehat{\boldsymbol{\theta}}(k-1) + \epsilon E[P(i)\boldsymbol{\phi}(i)(y(i) - \boldsymbol{\phi}^T(i)\widehat{\boldsymbol{\theta}}(i-1)]$, where the incremental term $\epsilon E[P(i)\boldsymbol{\phi}(i)(y(i) - \boldsymbol{\phi}^T(i)\widehat{\boldsymbol{\theta}}(i-1))]$ is the arithmetic mean of previous corrections.

4.3 Optimization Objective: the Game-based Objective Function

We have designed the payoff function following the principles of the Cournot Oligopoly game. We propose an optimal control solution to the game, which takes this payoff function as the *objective function* in the control system. Due to the unfeasibility for user i to observe $s^{-i}(k)$ and $c^{-i}(k)$ (defined previously in Table 1) at the time of decision making, the objective function (we again use $u_i(k)$ as a convenience) may be defined as follows, which is slightly different from the payoff function in Eq. (5):

$$u_i(k) = c_i(k)\cdot\{\lambda[\sum_{j\neq i} s_j(k-1) + s_i(k)] - [\sum_{j\neq i} c_j(k-1) + c_i(k)] - \frac{b_i(k)}{B_i(k)}\cdot e^{\frac{b_i(k)}{B_i(k)}}\}. \quad (8)$$

In Eq. (8), however, the terms $\sum_{j\neq i} s_j(k-1)$ and $\sum_{j\neq i} c_j(k-1)$ are global information and may not be conveniently known or observable to the peer users. Within a small-scale peer-to-peer group, if the decision updating period is sufficiently long, it is feasible for each peer user to constantly observe the $s_j(k)$, $c_j(k)$ and $b_j(k)$ values of all other peers. Thus, an arbitrary peer user i may directly use Eq. (8) as its decision making criterion. In this case, each peer user always attempts to form its best strategy based on the strategies selected by other peers in the previous stage, and users sequentially make their decisions in an iterative manner. If the iteration converges, it must converge to the Nash equilibrium.

As the network becomes larger, collecting complete information of all peers becomes infeasible. In such cases, certain characteristics of the peer-to-peer application may be of assistance in *estimating* such global information. In peer-to-peer storage systems, for example, if we assume that the location and lookup of data blocks are based on distributed hash [1, 3], there is a high probability that ideal load balancing is achieved. In this case, if the user knows the total number of peers in the system (N), it may use $(N-1) \cdot s_i(k)$ to estimate $\sum_{j \neq i} s_j(k)$. Such an estimation is not as accurate, but it eliminates the message passing overhead of exchanging peer information.

As the network scales further up (*i.e.*, large-scale network), it is increasingly difficult for each peer user to even have the knowledge of the total number of users. In such cases, we modify the objective function as follows, so that users only rely on locally observable parameters to make the decision (note that this extension has deviated from the game setting):

$$u_i(k) = c_i(k) \cdot [\lambda s_i(k) - c_i(k) - \frac{b_i(k)}{B_i(k)} e^{b_i(k)/B_i(k)}] \tag{9}$$

4.4 The Optimal Control System

Our game-based optimal contribution mechanism periodically calculates $c_i(k)$ that maximizes $u_i(k)$, subject to the estimated system behavior (the identified plant model) and the upper bounds of $c_i(k)$ and $b_i(k)$.

Assume all users have the full knowledge of $c_i(k)$ and $s_i(k)$ of one another, the optimal control problem can be formulated as follows:

$$\begin{cases} c_i^*(k) = \arg\max c_i(k)\{\lambda[\sum_{j\neq i} s_j(k-1) + s_i(k)] - [\sum_{j\neq i} c_j(k-1) + c_i(k)] \\ \qquad\quad -\frac{b_i(k)}{B_i(k)} e^{b_i(k)/B_i(k)}\} & (10.1) \\ c_i(k) \leq C_i(k) & (10.2) \\ b_i(k) \leq B_i(k) & (10.3) \\ s_i(k) = [1-\widehat{\beta}_i(k-1)]s_i(k-1) + \widehat{\alpha}_i(k-1)[c_i(k) - s_i(k-1)] + t_i(k) & (10.4) \\ b_i(k) = \sum_{j=0}^{n} \widehat{\gamma}_{i,j}(k-1) s_i(k-j) + w_i(k) & (10.5) \end{cases} \tag{10}$$

$t_i(k)$ and $w_i(k)$ are noises that may not be observed. We thus approximate them with the *estimation errors* of $s_i(k)$ and $b_i(k)$: $t_i(k) \doteq s_i(k) - \widehat{\alpha_i}(k-1)[c_i(k) - s_i(k-1)] + [\widehat{\beta}_i(k-1) - 1]s_i(k-1)$, $w_i(k) \doteq b_i(k) - \sum_{j=0}^{n} \widehat{\gamma}_i(k-1)s_i(k-j)$. Since they still cannot be evaluated as $c_i(k)$, $s_i(k)$ and $b_i(k)$ are unknown for the moment, we replace them with the average values of recent errors, *i.e.*, $t_i(k) \doteq E(t_i)$, and $w_i(k) \doteq E(w_i)$.

Substitute (10.3) to (10.5), then substitute (10.5) to (10.4), we may obtain another upper bound on $c_i(k)$:

$$\begin{aligned} c_i(k) \leq \bar{c}_i(k) = \tfrac{1}{\widehat{\alpha}_i(k-1)}\{ & \tfrac{1}{\widehat{\gamma}_{i,0}(k-1)}[B_i(k) - w_i(k) - \textstyle\sum_{j=1}^{n} \widehat{\gamma}_{i,j}(k-1)s_i(k-j)] \\ & +[\widehat{\alpha}_i(k-1) + \widehat{\beta}_i(k-1) - 1]s_i(k-1) - t_i(k)\} \end{aligned}$$

Combining it with (10.2), we obtain the tight upper bound on $c_i(k)$:

$c_i(k) \leq \min\{\bar{c}_i(k), C_i(k)\} = \widehat{c}_i(k)$.

Hence, the original problem is transformed to

$$\begin{cases} c_i^*(k) = \arg\max u_i(c_i(k)) = \arg\max c_i(k)\{\lambda[\sum_{j\neq i} s_j(k-1) + s_i(k)] \\ \qquad -[\sum_{j\neq i} c_j(k-1) + c_i(k)] - \frac{b_i(k)}{B_i(k)} e^{b_i(k)/B_i(k)}\} \\ c_i(k) \leq \widehat{c}_i(k) \end{cases}$$

Similarly, in the case of large-scale networks, the optimal control problem is the following:

$$\begin{cases} c_i^*(k) = \arg\max u_i(c_i(k)) = \arg\max c_i(k)[\lambda s_i(k) - c_i(k) - \frac{b_i(k)}{B_i(k)} e^{\frac{b_i(k)}{B_i(k)}}] \\ c_i(k) \leq \widehat{c}_i(k) \end{cases}$$

It can be shown that, in both cases, $\frac{\partial^2 u_i(c_i(k))}{\partial c_i(k)^2}$ may not be negative definite (depending on the estimated parameters of the plant), thus, the solution to the above problem should be chosen as $c_i^*(k) = \arg\min u_i(c_i(k)), c_i(k) \in \{\widehat{c}_i(k), \{u_i(\widetilde{c}_i(k))\}\}$, where $\{u_i(\widetilde{c}_i(k))\}$ are the local maximums or minimums of $u_i(c_i(k))$ that satisfy the first order condition of the objective function.

By reforming the optimal control problem at every stage of the game, the best contribution strategy of user i with respect to other users' decisions can be derived as the *optimal control law* to the dynamic system. In the example of peer-to-peer storage systems, the local controller periodically performs the following tasks:

(1) *Local observations.* Before the control decision is made for the kth time slot, the upper bound of storage contribution $C_i(k)$ and the upper bound for bandwidth contribution $B_i(k)$ should be determined at the beginning of slot k. The actual usage of storage $s_i(k-1)$ and bandwidth consumption $b_i(k-1)$ must be measured and calculated at the end of time slot $(k-1)$.

(2) *Information exchange.* Depending on different assumptions with respect to the scale of the peer-to-peer network, peers may need to exchange their local observations $s_i(k-1)$ and $c_i(k-1)$ with other peers.

(3) *System identification.* Due to the time-variant and stochastic nature of the system view, the model of the plant needs to be periodically estimated based on the locally observed values of $c_i(k)$, $s_i(k)$, and $b_i(k)$.

(4) *Constrained optimization.* Besides the upper bounds on $c_i(k)$ and $b_i(k)$, the acceptable user strategies are further constrained by the system behavior predicted from the estimated system model. Thus, the decision drawn will be optimal in terms of actual system performance (*e.g.*, bandwidth consumption).

5 Performance Evaluation

We perform simulations to compare the performance of two categories of solutions to the incentive engineering problem in peer-to-peer systems: the proposed optimal control based solution (referred to as *solution* 1 henceforth) and the more primitive game-theoretic solution (referred to as *solution* 2). In the following presentation, emphasis is placed on revealing the fundamental differences between the two types of decision making processes. Again, we take the peer-to-peer storage system as a case study in all our evaluations.

5.1 Simulation Settings

In all the experiments, we take 50 peers with heterogeneous but constant upper bounds, C_i and B_i, on storage contributions and bandwidth consumptions. Periodically, new data insertion requests for the entire system are generated according to a sine function. At different peers, the amounts of insertion requests are approximately proportional to their contributed but unused storage space. Deletion operations are generated independently for individual peers, which are in approximate proportion to their contributed and used space.

The plant model is assumed as follows:

$$\begin{cases} s_i(k) = [1-\beta_i(k)]s_i(k-1) + \alpha_i[c_i(k) - s_i(k-1)] + t_i(k) \\ b_i(k) = s_i(k) + 0.5s_i(k-1) + w_i(k) \end{cases}$$

where $\beta_i(k)$ stands for the deletion rate that occurs at user i in slot k, which is a white noise with mean in $[0,1]$ and variance 1; $\alpha_i(k)$ corresponds to the real insertion rate seen by peer i, which is affected by the total amount of data insertion requests and the total unused storage in the system, and the current contribution amount $c_i(k)$ of user i; $t_i(k)$ and $w_i(k)$ are zero-mean white noises, representing the uncertain factors in the external system, with regard to data insertion, deletion and bandwidth consumption.

We assume that the decision updating period is sufficiently long, so that peers exchange local observations on $s_i(k)$ and $c_i(k)$ to assist decision making. Thus, in both approaches to be evaluated, the objective function (in *solution* 1) and the payoff function (in *solution* 2) are periodically updated according to Eq. (8).

Solution 2 reaches the result by solving the following optimization problem:

$$\begin{cases} c_i^*(k) = \arg\max c_i(k)\{\lambda \sum_j s_j(k-1) - [\sum_{j\neq i} c_j(k-1) + c_i(k)] \\ \qquad\qquad -\frac{b_i(k-1)}{B_i(k)} e^{b_i(k-1)/B_i(k)}\} \\ c_i(k) \le C_i(k) \end{cases} \tag{11}$$

where the old values of $s_i(k-1)$ and $b_i(k-1)$ are used to form the optimization goal, and a single scalar $c_i^*(k)$ is derived as the optimal decision on $c_i(k)$.

Solution 1, instead, relaxes $s_i(k)$ and $b_i(k)$ to be unknown, and employs three additional constraints (the estimated system equations and the bandwidth upper bound) to the optimization problem as in Eq. (10). Therefore, $c_i^*(k)$ is determined along with the estimates of $s_i(k)$ and $b_i(k)$.

5.2 Experimental Results

Since *solution* 1 relies on system identification to restrict the acceptable solution set (strategy space), the correctness of the estimated parameters directly affects the user's final decision. As Fig. 6 has shown, our parameter estimation procedure gives satisfactory estimates.

As will be evident in forthcoming results, in *solution* 2, user decisions fluctuate evidently under system dynamics, so does the entire system capacity (Fig. 7(A)). The reason is that, in *solution* 2, the cost term in the payoff function is evaluated at the bandwidth consumption for the previous time slot, thus, it forms a strong negative feedback for the contribution decision in the next slot. Since users make decisions based on their *static* views (*i.e.*, observed values of $s_i(k)$, $c_i(k)$, and $b_i(k)$) of the underlying dynamic system, without further knowledge of its future variations, they tend to make decisions that maximize the current payoff function, but stimulate higher bandwidth consumption, equivalently, higher costs in the payoff function, for the coming slot. Hence, the decision on contribution for the next slot may drop steeply, and further induce lower costs subsequently.

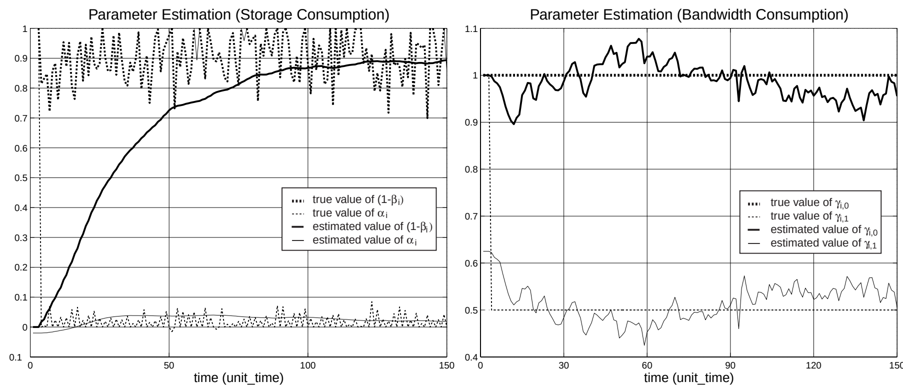

Fig. 6. Results of system identification. (A) Coefficients of the storage dynamic equation. (B) Coefficients of the bandwidth dynamic equation.

System Capacity The total storage space provided by the system determines to a large extent the benefit a user can receive. We use the notion *equivalent data insertion requests*, which equals to the amount of insertion requests subtracted by the amount of deleted data, to depict the variation of storage requirement in the system. As Fig. 7 has shown, both solutions render adaptable system capacity in face of system dynamics (stimulated by the equivalent data insertion requests), but *solution* 2 reacts more sensitively to the system variations. Besides, although decisions in *solution* 2 fluctuate more heavily, the average system capacity is higher than that achieved in *solution* 1. Both the significant capacity fluctuation and the augmented capacity are results of strong feedbacks between the bandwidth consumption and the contribution decision, which come with remarkable costs.

Bandwidth Stress As Fig. 7(B) shows, bandwidth consumption in *solution* 2 may consistently exceed the prescribed upper bound; the quality of service that individual peer users receive from the system, and that provided to the other peer users, may be severely degraded as a result.

The reason is that, *solution* 2 does not explicitly consider the effects that each contribution decision has on the observable system status ($s_i(k)$ and $b_i(k)$), it tends to degrade network performance by more aggressively consuming user bandwidth. Primitive game-based strategies cannot avoid such phenomena, since the cost term in the payoff function only serves as a *virtual* penalty on the bandwidth consumption of the previous period: contribution quantities that may significantly increase future bandwidth

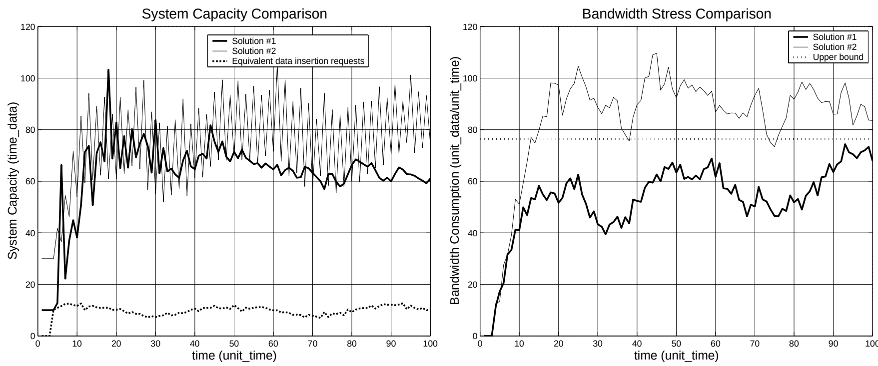

Fig. 7. Comparisons between the two solutions. (A) System capacity; (B) Bandwidth stress.

consumption are still acceptable within the context, as long as the payoff function is maximized. On the contrary, the optimal control based solution effectively alleviates such deterioration on system performance, by restricting user strategy space with the estimated system model, so that the contribution decision is derived as an attempt to maximize the control objective, subject to the constraint of possible system behavior.

Storage Utilization Fig. 8(A) shows the variations of storage usage for both solutions, and Fig. 8(B) illustrates the corresponding storage utilization factors. It can be seen that, the storage utilization is relatively stable for *solution* 1, due to its smoother contribution variations. In addition, we set λ to $1.25(N+1)/N$ in our simulation, which corresponds to an expected storage utilization of 80%, and it approximately agrees with our simulation results.

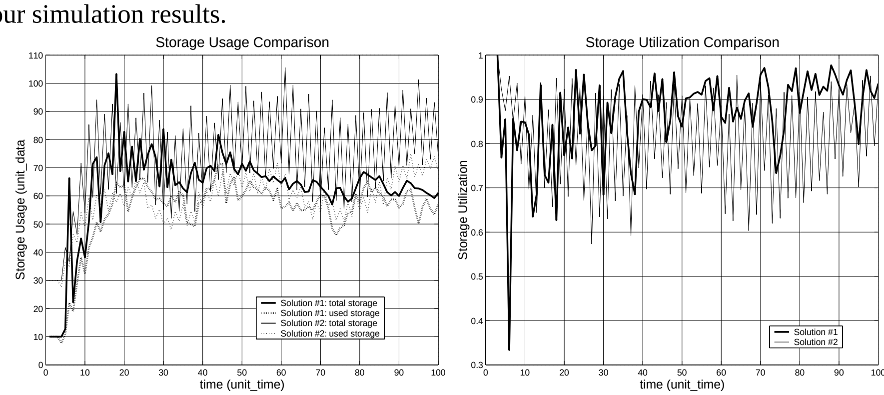

Fig. 8. Comparisons between the two solutions. (A) Contributed and used storage; (B) Capacity utilization.

In summary, we have verified that, in peer-to-peer applications, primitive game-based strategies may not be readily applicable to manipulate real-time systems, due to their inherent restrictions of considering the underlying system with a static view. The limitation is alleviated when the physical rules governing the dynamic system behavior are explicitly incorporated into the decision making procedure. From an optimal control perspective, the user strategy space can be constrained by feasible system behavior, and incentives can be more adequately engineered by properly designing payoff functions. Therefore, distributed users may spontaneously make decisions to benefit themselves and other users, unconsciously assisting the maintenance of optimal states in the global peer-to-peer system.

6 Related Work

We have resorted to two theoretical tools, game theory and control theory, to address the problem of incentive engineering. Both theories have been extensively employed in various network-related areas, but their applications to peer-to-peer system have only recently begun to be studied.

As a powerful tool in solving multi-person decision making problems, game theory has been applied in multiple channel access arbitration [10], optimal routing [11–13], as well as flow and congestion control [14]. Some of the recent work has started to apply game theory in peer-to-peer applications. Golle *et al.* [15], for instance, have discussed game theory applications in peer-to-peer file sharing applications with centralized servers. In particular, they take into account the effect of users' levels of altruism on their behavior of contributing and receiving, and construct the user strategy space accordingly. Liao *et al.* [16] have focused on incentive provisioning for wireless access services, and constrained user strategy spaces with service purchasing power and price-service menu, so that only desirable cooperative behavior is permitted. Feigenbaum *et al.* [8] have studied the more general problem of mechanism design, which encourages users to behave in the way that leads to the system desirable outcome, by properly designing associated payoffs and specifications that are computationally tractable.

For the purpose of providing reasonable incentives and achieving desirable system performance in peer-to-peer applications, we believe that to investigate the relationship between user behavior and system performance, which closely depends on the mathematical model of the underlying system, is the primary task we should undertake. Only based on adequate knowledge about the relationship, can we proceed to design satisfactory mechanisms to control user behavior. Control theory, fortunately, provides the right concepts and techniques for modeling system dynamics, analyzing performance, and designing appropriate controllers to regulate system behavior. We believe that employing control-theoretic techniques in peer-to-peer networks will provide a solid basis for incentive engineering research, and will generate effective solutions. We have not been aware of any existing work that takes a control-theoretic approach to address problems in this aspect.

7 Conclusions

In this paper, we have investigated the issue of peer-to-peer incentive engineering from both the game-theoretic and the control-theoretic perspective. The original contributions of this paper are two-fold. First, we model the situation as a game and propose to *design* the payoff function such that, as the users maximize their own net gains, the desirable global system performance is obtained. Second, we propose an optimal control solution to regulate users' adaptive behavior of the game, by using the designed payoff function directly as the objective function that the controller optimizes. Our experimental results have demonstrated that the control solution behaves more stably in face of system dynamics, while the primitive game solution may also achieve acceptable performance in most cases. To the best of our knowledge, there do not exist previous studies in the area of incentive engineering based on either the design of payoff functions in a game (*i.e.*, the principles of inverse games) to achieve global properties, or the use of an optimal control system to adapt to system dynamics while catering to the selfishness of users.

References

1. F. Dabek, M. F. Kaashoek, D. Karger, R. Morris, and I. Stoica, "Wide-Area Cooperative Storage with CFS," in *18th ACM Symposium on Operating Systems Principles (SOSP 2001)*, October 2001.
2. J. Kubiatowicz, D. Binderl, Y. Chen, and S. Czerwinskl, "OceanStore: An Architecture for Global-Scale Persistent Storage," in *The Ninth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2000.
3. A. Rowstron and R. Druschel, "Storage Management and Caching in PAST, a Large-Scale Persistent Peer-to-Peer Storage Utility," in *18th ACM Symposium on Operating Systems Principles (SOSP 2001)*, October 2001.
4. D. G. Andersen, H. Balakrishnan, M. Frans Kaashoek, and R. Morris, "Resilient Overlay Networks," in *18th ACM Symposium on Operating Systems Principles (SOSP 2001)*, October 2001.
5. I. Foster, "Internet Computing and the Emerging Grid," *Nature*, December 2000.
6. S. Saroiu, P. Gnummadi, and S. Gribble, "A Measurement Study of Peer-to-Peer File Sharing Systems," in *Proc. of SPIE/ACM Conference on Multimedia Computing and Networking (MMCN 2002)*, 2002.
7. E. Adar and B. Huberman, "Free Riding on Gnutella," *First Monday*, vol. 5, no. 10, 2000.
8. J. Feigenbaum and S. Shenker, "Distributed Algorithmic Mechanism Design: Recent Results and Future Directions," in *Sixth International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (Dial'M 2002)*, September 2002.
9. K. J. Astrom and B. Wittenmark, *Adaptive Control, 2nd Edition*, Addison-Wesley Publishing Company, 1995.
10. A. B. MacKenzie and S. B. Wicker, "Selfish Users in Aloha: A Game-Theoretic Approach," in *Proceedings of IEEE Vehicular Technology Conference*, 2001.
11. A. Orda, R. Rom, and N. Shimkin, "Competitive Routing in Multi-User Communication Networks," *IEEE/ACM Transactions on Networking*, vol. 1, pp. 510 – 521, 1993.
12. Y. A. Korlis, A. A. Lazar, and A. Orda, "Architecting Noncooperative Networks," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 8, 1995.
13. R. J. La and V. Anantharam, "Optimal Routing Control: Game Theoretic Approach," in *Proceedings of the 36th Conference on Decision and Control*, 1997.
14. T. Alpcan and T. Basar, "A Utility-Based Congestion Control Scheme for Internet-Style Networks with Delay," submitted to IEEE Infocom 2003.
15. P. Golle, K. L. Brown, I. Mironov, and M. Lillibridge, "Incentives for Sharing in Peer-to-Peer Networks," in *Proceedings of the 2nd International Workshop on Electronic Commerce*, 2001.
16. R. R.-F. Liao, R. H. Wouhaybi, and A. T. Campbell, "Incentive Engineering in Wireless LAN Based Access Networks," in *Tenth International Conference on Network Protocols*, November 2002.