

# P2P Trading in Online Social Networks: The Value of Staying Connected

Zhengye Liu<sup>†</sup>, Hao Hu<sup>‡</sup>, Yong Liu<sup>‡</sup>, Keith Ross<sup>†</sup>, Yao Wang<sup>‡</sup>, and Markus Mobius<sup>\*</sup>

<sup>†</sup>Computer Science and Engineering, Polytechnic Institute of NYU

<sup>‡</sup>Electrical & Computer Engineering, Polytechnic Institute of NYU

<sup>\*</sup>Economics, Harvard University

**Abstract**—The success of future P2P applications ultimately depends on whether users will contribute their bandwidth, CPU and storage resources to a larger community. In this paper, we propose a new incentive paradigm, *Networked Asynchronous Bilateral Trading (NABT)*, which can be applied to a broad range of P2P applications. In NABT, peers belong to an underlying online social network (such as Facebook), and each pair of friends keeps track of a credit balance between them. When user Alice provides a service (a file, storage space, computation and so on) to her friend Bob, she charges Bob credits. Thus, in NABT, there is no global currency; instead, there are only credit balances maintained between pairs of friends. NABT allows peers to supply each other asynchronously and further allows peers to trade with remote peers through intermediaries. We theoretically show that NABT is perfectly efficient with balanced demands and supports “networked tit-for-tat”. The efficiency of NABT with unbalanced demands is determined by the min-cut of credit limits of the underlying social network. Using simulations driven by MySpace traces, we demonstrate that a simple two-hop NABT design can have high trading efficiency, provide service differentiation, exploit trading intermediaries, and discourage free-riders.

## I. INTRODUCTION

Although P2P has proven itself as a viable architectural paradigm for a variety of large-scale distributed applications, P2P is far from reaching its full potential. Peers possess surplus bandwidth, storage and CPU resources, with the surplus fluctuating throughout the day. When aggregated together across all peers worldwide, these unused resources constitute a huge, untapped resource pool. The success of future P2P applications ultimately depends on whether users contribute their resources to a larger community, which is the challenge of *incentive design*.

A natural approach to providing incentives is to use a single *global currency*, whereby peers earn currency units when contributing resources. A global currency system mimics real-world currency and transactions, where each user has a savings account that is decremented when it consumes services and incremented when it supplies services. Users’ savings could be tracked using a centralized bank or using some form of digital cash [5]. From economic theory, we know that global currency systems are highly efficient [18]. Global currencies, however, require a high degree of coordination: central banks rely on a well-functioning legal system that enforces contracts, punishes counterfeiters and resolves disputes. Moreover, central banks need to build a reputation against creating unlimited amounts

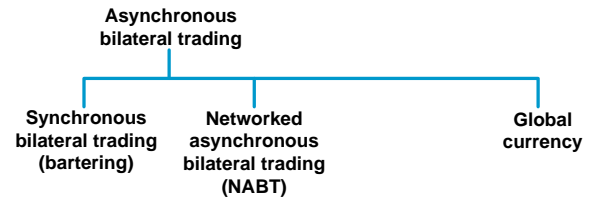


Fig. 1. The spectrum of economic systems.

of currency which would ignite inflation and undermine the usage of currency. Although there have been several proposals for using global currency in P2P [23], [22], [2], and several systems have actually been built, there has not been a deployment to date that has taken hold on a large scale [5]. There has also been numerous proposals for global P2P reputation systems [13], [11]; but to date there is not a large-scale P2P deployment that has successfully used reputation.

As illustrated in Figure 1, global currency is at one extreme of the spectrum of economic systems. At the other extreme of the spectrum is *barter*, for which peers *bilaterally* and *synchronously* trade resources. Barter is appealing because of its extreme simplicity, as there is no need for currency, banks, public-key infrastructures, elaborate dispute-resolution mechanisms or bank regulation. BitTorrent, a highly-successful P2P file sharing protocol, employs a bartering-like scheme called “tit-for-tat.” In BitTorrent’s tit-for-tat, Alice gives more chunks to those peers who are currently giving her the most chunks [6]. This trading is *synchronous* because it is taking place within very short time periods, on the order of tens of seconds. But BitTorrent’s bartering scheme is only “partially” successful in that it can be circumvented [15], [16], [19], [20]. More importantly, it is difficult to trade long-tail content with BitTorrent because users’ interests in long-tail content are rarely synchronously matched, and there is no incentive for a peer to contribute if it has no immediate demand.

In the spectrum of economic systems as illustrated in Figure 1, ranging from bilateral trading at one extreme to a global currency at the other, the sweet spot for P2P incentive design likely lies somewhere in-between. This paper explores a new paradigm for incentives, *Networked Asynchronous Bilateral Trading (NABT)*. In NABT, peers belong to an underlying online social network (such as Facebook), with each peer having a set of friends. In NABT peers can trade asynchronously –

e.g., one day Alice can perform computation for Bob, and on another day Bob can reciprocate by supplying a file to Alice. To incentivize friends to help each other, each pair of friends maintains a credit balance and a credit limit. In this manner, Alice will want to provide service to Bob so that she in turn can gain service from him in the future. NABT further increases the market efficiency by allowing trades to pass through intermediaries. For example, if Alice and Charlie have a common friend, Bob, then Charlie can serve Alice through Bob. In the process, Bob will decrease his debt with Alice but increase his debt with Charlie. This mechanism makes Alice’s credit with Bob more valuable because she can use that credit to trade both directly with Bob and indirectly with Bob’s friends. In this paper we make the following contributions:

- We introduce NABT as a new P2P trading paradigm for online social networks. NABT solves two fundamental problems of the traditional bartering: asynchronicity over time and asynchronicity over nodes.
- We develop mathematical models to study the fundamental efficiency problem of NABT. We theoretically show that NABT is perfectly efficient with balanced demands and supports “networked tit-for-tat”. The efficiency of NABT with unbalanced demands is determined by the min-cut between service sources and service sinks. We further establish the memoryless property of dynamic credit transfer routing in NABT.
- We propose a simple approach for dispute resolution. Specifically, to handle cheaters that do not provide server after accepting credit transfer, we design an Additive-Increase Multiplicative Decrease credit limit algorithm. Through simulation, we show that this mechanism effectively isolates cheaters without punishing cooperative peers.
- Using extensive simulations, we illustrate NABT within the context of P2P file sharing over an online social network. The trace is driven by a large-scale MySpace trace. We demonstrate that a simple *two-hop* NABT design can have high trading efficiency (close to that of global currency), provide service differentiation, exploit trading intermediaries, and discourage free-riders.

The rest of the paper is organized as follows. In Section II, we introduce NABT as a new P2P incentive paradigm in online social networks. We mathematically analyze NABT’s efficiency in Section III. In Section IV, we evaluate the performance of two-hop NABT in P2P file sharing through simulations driven by the MySpace trace. We summarize the related work on P2P incentives in Section V and conclude in Section VI.

## II. NABT IN ONLINE SOCIAL NETWORKS: A NEW P2P INCENTIVE PARADIGM

Most P2P resource markets involve long-tail content and diverse service interest, synchronicity is highly constraining and inefficient. Synchronous trading fails along two dimensions: (i) demand requests occur at different times (*asynchronicity*

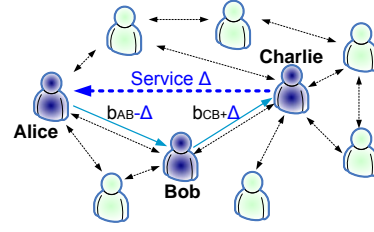


Fig. 2. Example of Credit Transfer in Social Network

*across time*) and (ii) users demand and supply are not typically bilaterally matched over time (*asynchronicity across nodes*).

In our proposed Networked Asynchronous Bilateral Trading (NABT) paradigm, we use credits to solve the problem of asynchronicity across time, while using intermediaries to solve the problem of asynchronicity across nodes. The basic elements of NABT are as follows:

- **Online social network.** Peers belong to an underlying online social network, such as Facebook or MySpace. Each peer in the social network has a set of friends.
- **Credit limits.** Each peer  $i$  sets a credit limit  $C_{ij}$  for each friend  $j$ . The magnitude of  $C_{ij}$  quantifies the trust between the two friends and depends on their past trading history<sup>1</sup>.
- **Credit balance.** At any given time there is a credit balance between friends. Let  $b_{ij}$  denote the amount of credits user  $j$  owes user  $i$ . By definition, we have  $b_{ij} = -b_{ji}$ . Because of the credit limits, the credit balance satisfies

$$-C_{ji} \leq b_{ij} \leq C_{ij}. \quad (1)$$

- **Asynchronous trading.** When user Alice provides a good or service (a file, storage space, computation, expert advice and so on) to Bob, she charges Bob a certain number of credits. Her credit balance with Bob then increases correspondingly. Alice will not provide the service if Bob’s resulting debt would exceed the credit limit  $C_{ij}$ .
- **Trading via intermediaries.** Suppose Charlie wants a good or service from Alice, but he is not one of her direct friends. If they both have a common friend, say Bob, then Charlie can still obtain service from Alice using their mutual friend as a credit intermediary. In the process, Bob will decrease his debt with Charlie but increase his debt with Alice, see Figure 2. In any transaction between Alice and Charlie, there can be multiple intermediaries and, in fact, multiple paths of intermediaries.

Unlike a currency scheme, NABT does not involve a global currency infrastructure, and is void of banks, public-key infrastructures, elaborate dispute-resolution mechanisms and bank regulation. Unlike BitTorrent, the scheme allows for asynchronous trading, and trading of mixed services.

<sup>1</sup>In practice, the credit limit  $C_{ij}$  would not be explicitly set by user  $i$ , but instead by a local software agent acting on user  $i$ ’s behalf. The users may configure the agents with policies, providing guidelines to the agent, which may adaptively modify its credit limits.

Importantly, NABT exploits the built-in human-established trust relationships among friends in online social networks – it is anticipated that a user in an online social network has much less motivation to cheat resources from his/her real world friends. Indeed, the built-in trustworthiness in online social networks has been widely studied to enhance security in computer networks [3], [21], [24].

### III. EFFICIENCY OF NABT

In a global currency system, if Alice has enough money, she can always buy a service directly from other peers using the global currency. However, in NABT, even if Alice has sufficient pair-wise credits with her friends, she may not be able to pass these credits to the service suppliers, due to the credit limits at the intermediaries. In this section, we analytically address the most fundamental question – *how efficient is NABT?*

#### A. NABT Efficiency Model

We consider a set  $U$  of users connected in a social network  $G_S = (U, F)$ , where the social link set  $F \subset U \times U$  defines the friends relations between users. Since friendship is generally a mutual relationship, there exist a pair of directed links between two friends. Suppose each user can provide some services, and also wants to consume some services provided by other users. We first consider a model for *static demand* where all users post their demands in the same time slot. In Subsection III-F we will extend the model to cover dynamic demand, where users introduce new demands from time slot to the next.

We can define a demand graph  $G_D = (U, D)$ , where  $D \subset U \times U$  is the service demand profile among users. For a service demand  $d = (k, l) \in D$ ,  $k$  is the provider and  $l$  is the consumer of the service. User  $k$  charges user  $l$  a cost of  $h^{(k,l)}$  for providing the service. We define the associated  $|U| \times |U|$  demand matrix  $H = [H(k, l)]$  as  $H(k, l) = h^{(k,l)}$  if  $(k, l) \in D$ , and  $H(k, l) = 0$  if  $(k, l) \notin D$ .

With NABT, friends can exchange services asynchronously as the mutual credit limits allow. Initially, there is no credit balance between  $i$  and  $j$ ,  $b_{ij}(0) = b_{ji}(0) = 0$ . After  $i$  serves  $j$  once, the balance becomes  $b_{ij}(1) = h^{(i,j)}$ . If the next time  $i$  obtains a service  $(j, i)$  from  $j$ , the balance becomes  $b_{ij}(2) = b_{ij}(1) - h^{(j,i)} = h^{(i,j)} - h^{(j,i)}$ . A new service transaction between  $i$  and  $j$  can be admitted if and only if the resulting credit balance  $b_{ij}$  after the new transaction meets the credit balance constraints between  $i$  and  $j$  as summarized in (1).

To facilitate trading via intermediaries, we introduce a credit transfer mechanism through a path of friend links. Specifically, if  $l$  wants to obtain some service provided by  $k$ ,  $l$  first tries to find a path from  $k$  to  $l$  in the social network,  $p(k, l) = \{k = r_0 \rightarrow r_1 \rightarrow r_2 \rightarrow \dots \rightarrow r_{m-1} \rightarrow r_m = l\}$ . Since friend links are bi-directional,  $l$  then initiates a sequence of credit transfers in the reverse direction:  $r_m$  pays  $h^{(k,l)}$  credits to  $r_{m-1}$ ,  $n = m, \dots, 1$ . After the credit transfers,  $k$  can provide the service to  $l$  and the credit balance on each node is updated<sup>2</sup>.

<sup>2</sup>Pairwise credits between different friend pairs are not exchangeable. Therefore, the credit balances between different friend pairs cannot be merged.

#### B. Credit Transfer Routing in NABT

More generally, a service demand (e.g., a file transfer or a computation) can be served as long as a legitimate credit transfer flow can be established from the provider of the service to the consumer of the service. We characterize the credit transfer on a social link  $\langle i, j \rangle$  for demand  $d$  using variable  $x_{\langle i, j \rangle}^d$ , which is defined as the amount of credits that node  $j$  takes from node  $i$  for service demand  $d$ .  $\{x_{\langle i, j \rangle}^d, \langle i, j \rangle \in F\}$  should satisfy *flow conservation* on all nodes in the network:

$$\sum_{i: \langle i, u \rangle \in F} x_{\langle i, u \rangle}^d - \sum_{j: \langle u, j \rangle \in F} x_{\langle u, j \rangle}^d = \begin{cases} -h^d & \text{if } u = p(d) \\ h^d & \text{if } u = c(d) \\ 0 & \text{otherwise} \end{cases}, \quad (2)$$

where  $p(d)$  and  $c(d)$  denote the provider and consumer for service  $d$  respectively,  $\forall u \in U$  and  $\forall d \in D$ .

When there are multiple simultaneous service demands, the total aggregate credit transfers on link  $\langle i, j \rangle$  and  $\langle j, i \rangle$  can be calculated as  $\sum_{d \in D} x_{\langle i, j \rangle}^d$  and  $\sum_{d \in D} x_{\langle j, i \rangle}^d$  respectively. Therefore the resulted credit balance between user  $i$  and  $j$  after all credit transfers is  $b_{ij} = \sum_{d \in D} (x_{\langle i, j \rangle}^d - x_{\langle j, i \rangle}^d)$ . We have to make sure that the resulted credit balances on all links after executing all services are bounded by their credit limits:

$$-C_{ji} \leq \sum_{d \in D} (x_{\langle i, j \rangle}^d - x_{\langle j, i \rangle}^d) \leq C_{ij}, \quad \forall \langle i, j \rangle \in F. \quad (3)$$

#### C. Formal Problem Statement

Given a social network  $S = (U, F)$  with links weighted by credit limits  $\{C_{ij}, \langle i, j \rangle \in F\}$ , and the set of service demands  $D$  with the associated demand matrix  $H$ , the credit transfer routing problem is to find a set of credit transfer flows  $\mathcal{X}(H) \triangleq \{x_{\langle i, j \rangle}^d, d \in D, \langle i, j \rangle \in F\}$  such that constraints defined in (2) and (3) are all satisfied. This problem is similar to the traffic routing problem in transport and computer networks if one views the credit limit as link capacity in a social network. One major difference is that the credit balance on a link can be negative and credit flows in opposite directions cancel each other.

#### D. Efficiency with Balanced Demand Set

In pure synchronous bilateral trade (barter), pairs of peers play tit-for-tat with each other and the pairwise service contribution and consumption balance out. In this subsection, we show that NABT supports *networked tit-for-tat*, that is, all demands can be met, if for each user, the total cost of service demands generated by the user equals to the total cost of service demands requested from the user. After establishing this result, in the next subsection we will consider the more general case of unbalanced demands.

*Definition 1:* Balanced Demand Matrix: a demand matrix  $H$  is called balanced if for each user, the total cost of service demands generated by the user equals to the total cost of service demands requested from the user, that is,

$$\sum_{l \in U} H(k, l) = \sum_{i \in U} H(i, k), \quad \forall k \in U$$

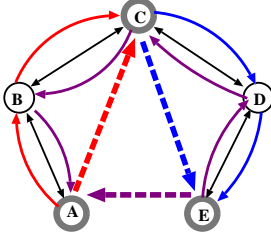


Fig. 3. Credit transfers for a balanced demand set

*Lemma 1:* If a set of demands form a loop in the demand graph  $G_D$  and the demand costs are the same, then all demands in the set can be executed simultaneously as long as users involved in the demand sets are connected in the social network  $G_S$ .

*Proof:* Let  $C = \{u_0 \rightarrow u_1 \rightarrow \dots \rightarrow u_m \rightarrow u_{m+1} = u_0\}$  be a loop in the demand graph, and  $h^{(u_n, u_{n+1})} = h, 0 \leq n \leq m$ . For  $0 \leq n \leq m-1$ , find a path  $P_n$  in  $G_S$  from  $u_n$  to  $u_{n+1}$ . Obviously  $P_{-m} \triangleq \bigcup_{0 \leq n \leq m-1} P_n$  is a path (might with loops) from  $u_0$  to  $u_m$ . Then  $P_m \triangleq \{\langle j, i \rangle \mid \text{the reverse link } \langle i, j \rangle \in P_{-m}\}$  forms a path from  $u_m$  to  $u_0$ . Now allocate credit flows for demands in set  $\{(u_n, u_{n+1}), 0 \leq n \leq m\}$  as follows:

$$\{x_{\langle i, j \rangle}^{(u_n, u_{n+1})} = h, \forall \langle i, j \rangle \in P_n, 0 \leq n \leq m\}$$

It can be easily verified that the flow conservation in (2) is maintained for all demands, and

$$\sum_{n=0}^m x_{\langle i, j \rangle}^{(u_n, u_{n+1})} = \sum_{n=0}^m x_{\langle j, i \rangle}^{(u_n, u_{n+1})} \quad \forall \langle i, j \rangle \in \bigcup_{0 \leq n \leq m} P_n.$$

In other words, the credit balance resulted from the executions of this set of service demands is zero on all social links involved. Therefore the credit balance constraints (3) are satisfied automatically.

In summary, a demand loop in the demand graph can be executed synchronously in a social network without leaving any balance on any social link. ■

As illustrated in Figure 3, five users form a social network with a chain topology:  $A \leftrightarrow B \leftrightarrow C \leftrightarrow D \leftrightarrow E$ . If C wants service from A, and E wants service from C, A wants service from E. Service providers and consumers are not directly connected in the social network. By setting up credit transfer path  $A \rightarrow B \rightarrow C$  for service (AC),  $C \rightarrow D \rightarrow E$  for service (CE), and path  $E \rightarrow D \rightarrow C \rightarrow B \rightarrow A$  for service (EA), all demands are executed, and the credit balances on all links remain zero.

*Lemma 2:* Any balanced demand matrix can be decomposed into finite demand loops in the demand graph.

*Proof:* In the demand graph  $G_D$ , assign the cost of a service demand as the weight of the corresponding demand link. For any node in the graph of a balanced demand set, the total weight on its ingress links equals to the total weight on its egress links. One can traverse the graph in the following way: starting from any node  $u_0$ , pick any egress link, say  $\langle u_0, u_1 \rangle$ ,

with positive weight, move to  $u_1$ ; since  $u_1$  has at least one positive weight ingress link, due to the balanced ingress and egress link weights, it must have at least one positive weight egress link, then pick any positive weight egress link of  $u_1$ , say  $\langle u_1, u_2 \rangle$ , move to  $u_2$ , using the same argument, the trip can continue until at some step  $n$ , the trip goes back to a previously visited node  $u_i, 0 \leq i < n$ , then we find a demand loop  $u_i \rightarrow \dots \rightarrow u_{n-1} \rightarrow u_n = u_i$ . Let  $h = \min_{i \leq l \leq n-1} h^{(u_i, u_{l+1})}$ , remove the identified loop from the graph by subtracting  $h$  from the weights of all links in the loop. The ingress and egress link weights on all nodes are still balanced after the loop removal. We can repeat the process until all link weights go to zero, or equivalently all demands have been decomposed into demand loops. ■

*Theorem 1:* Any balanced demand matrix can be executed simultaneously in a social network  $G_S$  as long as users involved in the demand sets are connected in  $G_S$ .

*Proof:* According to Lemma 2, we can decompose a balanced demand set into demand loops. According to Lemma 1, each demand loop can be executed sequentially without generating credit balance on any link. After all demand loops are executed, all demands in the original balanced set are executed. ■

Under balanced demand and simultaneous credit transfer, we therefore see that the NABT credit limits do not constrain network trading. This single slot version of NABT can be viewed as a generalization of the synchronous bilateral trading; however, *rather than playing tit-for-tat with a particular peer, peers in our P2P mechanisms effectively play tit-for-tat with the whole network.*

### E. Efficiency with Unbalanced Demand Set

In reality, service demands between peers are dynamic and unavoidably unbalanced at any given time instance. Under unbalanced demands, credit limits provide “cushion” to absorb the temporary service imbalance between peers. In this subsection, we quantify the levels of bilateral credits required to satisfy the demand when the service demands are unbalanced.

*Definition 2:* Unbalanced Demand Matrix: a demand matrix  $H$  is called unbalanced if there is at least one user such that the total cost of service demands generated by the user does not equal to the total cost of service demands requested from the user in the demand set:

$$\sum_{l \in U} H(k, l) \neq \sum_{i \in U} H(i, k), \exists k \in U$$

*Definition 3:* Reduced Demand Matrix: for an unbalanced demand matrix  $H$ , the reduced demand matrix  $\bar{H}$  is defined as  $H(k, l) = H(k, l) - \min(H(k, l), H(l, k))$ .

The reduced demand matrix captures the net demand between a pair of users. The following Lemma establishes the equivalence of the routing feasibility of a demand matrix and its reduced demand matrix.

*Lemma 3:* A demand matrix  $H$  is executable in a social network  $G_S$  if and only if the corresponding reduced demand matrix  $\bar{H}$  is executable in  $G_S$ .

*Proof:* **IF**  $\Leftarrow$ : If  $\bar{H}$  is executable, let  $\mathcal{X}(\bar{H})$  be the associated credit flow. Define  $\hat{H} = H - \bar{H}$ , where the subtraction is taken on each element.  $\hat{H}$  defines a balanced demand set. Due to Theorem 1, it is executable with some credit flow  $\mathcal{X}(\hat{H})$  with zero credit balance on all links. It can be easily verified that  $\mathcal{X}(\bar{H}) + \mathcal{X}(\hat{H})$  defines a legitimate credit flow for the original demand  $H = \bar{H} + \hat{H}$

**ONLY IF**  $\Rightarrow$ : If  $H$  is executable, let  $\mathcal{X}(H)$  be the credit flow. For each pair of users  $(k, l)$ , merge credit flows for two demands  $(k, l)$  and  $(l, k)$ . It can be easily verified that the combined flow implement the net demand between  $(k, l)$  (possibly with credit transfer loops). ■

In an unbalanced demand set, we can classify users based on their net service contribution to the whole system.

*Definition 4:* Service sources and sinks in unbalanced demand set: for a user  $u$  in an unbalanced demand set, calculate the difference between the cost of service that other users request from him and the cost of the service that he requests:  $w(u) = \sum_{l \in U} (H(u, l) - H(l, u))$ . If  $w(u) > 0$ , we call user  $u$  a service source, and  $p(u) = w(u)$  is its net service contribution; if  $w(u) < 0$ , we call user  $u$  a service sink, and  $n(u) = -w(u)$  is its net service consumption.

Note that  $\sum_{v \in U^-} n(v)$  is the *surplus demand aggregated across all service sinks*. Note that this critical quantity is zero for a balanced demand set but will be strictly positive for an unbalanced set. By definition  $\sum_{u \in U^+} p(u) = \sum_{v \in U^-} n(v)$  even for an unbalanced demand set. After classifying users into service sources and sinks, we can build an extended social network to study the efficiency of NABT.

*Definition 5:* For a social network  $G_S = (U, F)$  with an unbalanced demand set  $D$ , let  $U^+$  be the set of service sources, and  $U^-$  be the set of service sinks, we construct an extended social network  $G'_S = (U', F')$  as follows:

$$U' = U \cup \{s^+, s^-\}; \quad (4)$$

$$F' = F \cup \{\langle s^+, u \rangle\}_{u \in U^+} \cup \{\langle v, s^- \rangle\}_{v \in U^-}; \quad (5)$$

$$C_{\langle s^+, u \rangle} = p(u), \quad \forall u \in U^+; \quad (6)$$

$$C_{\langle v, s^- \rangle} = n(v), \quad \forall v \in U^-. \quad (7)$$

Figure 4 shows an extended social network for an original social network with ten users, where three users are service sources, three users are service sinks, four users are balanced. Now we are ready to present the main theorem that characterizes the efficiency of NABT with unbalanced demands.

*Theorem 2:* An unbalanced demand matrix  $D$  is executable in a social network  $G_S$  if and only if the min-cut between the source  $s^+$  and sink  $s^-$  in the extended social network  $G'_S$  is greater than or equal to  $\sum_{v \in U^-} n(v)$  (that is, greater than equal to the aggregate surplus demand).

*Proof:* **ONLY IF**  $\Rightarrow$ : In any credit transfer solution for  $D$ , the net credit flow from set  $U^+$  to  $U^-$  is  $M = \sum_{u \in U^+} p(u)$ . Therefore the min-cut in  $G'_S$  has to be at least  $M$ .

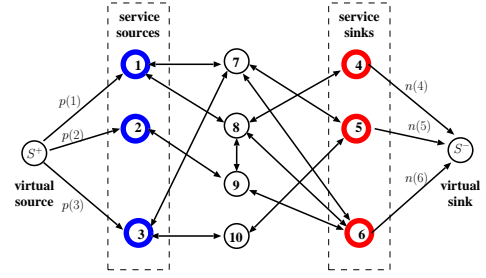


Fig. 4. Example of Extended Social Network

**IF**  $\Leftarrow$ : Given a min-cut between  $s^+$  and  $s^-$  in  $G'_S$  with size  $M$ , construct a flow from  $s^+$  and  $s^-$  with volume  $M = \sum_{u \in U^+} p(u)$ , due to the capacity constraint on link  $\langle s^+, u \rangle, u \in U^+$ , the flow routed from  $s^+$  to a service source  $u \in U^+$  is exactly  $p(u)$ , likewise, the flow routed from a service sink  $v \in U^-$  to  $s^-$  is exactly  $n(v)$ . On the virtual source node  $s^+$ , put different labels to flows distributed through different service source nodes. On each service sink node, calculates the volume of flows received from different service sources, denoted as  $f(u, v), u \in U^+, v \in U^-$ ,

$$\sum_{v \in U^-} f(u, v) = p(u), \quad \forall u \in U^+; \quad (8)$$

$$\sum_{u \in U^+} f(u, v) = n(v), \quad \forall v \in U^-. \quad (9)$$

Let's construct a new demand matrix  $\hat{H}$  as  $\hat{H}(u, v) = f(u, v)$  if  $u \in U^+$  and  $v \in U^-$ ; and  $\hat{H}(u, v) = 0$  otherwise. The maximum flow directly transfer credits for  $\hat{H}$  in  $G_S$ . So demands in  $\hat{H}$  can be synchronously executed in  $G_S$ . It can be easily verified that

$$\sum_{l \in U} (\hat{H}(k, l) - \hat{H}(l, k)) = \sum_{l \in U} (H(k, l) - H(l, k)), \quad \forall k \quad (10)$$

That is to say  $\hat{H}$  and  $H$  have exactly the same aggregate service unbalance distribution on each node.

Now construct another demand matrix  $\tilde{H}$  by "subtracting"  $\hat{H}$  from  $H$ . More specifically,

- 1)  $\tilde{H}(k, l) = H(k, l) - \hat{H}(k, l)$ ;
- 2)  $d(k, l) = \tilde{H}(k, l) - \tilde{H}(l, k)$ ;
- 3) if  $d(k, l) > 0$ , set  $\tilde{H}(k, l) = d(k, l)$  and  $\tilde{H}(l, k) = 0$ ; otherwise set  $\tilde{H}(k, l) = 0$  and  $\tilde{H}(l, k) = -d(k, l)$ .

Due to (10), it can be verified that  $\tilde{H}$  is a balanced demand set. Due to Theorem 1, we can find credit transfer flows for  $\tilde{H}$  without change the credit balance on links. Combine the credit flows for  $\tilde{H}$  and  $\hat{H}$ , we have a credit flow for  $\tilde{H} + \hat{H}$ . It can be checked that the reduced demand matrix for  $\tilde{H} + \hat{H}$  is the same as the reduced matrix for the original demand set  $H$ . Due to Lemma 3,  $H$  is also executable. ■

Theorem 2 extends the networked tit-for-tat results in Theorem 1. The efficiency of NABT is determined not by the service imbalance between pairs of users, only by the service imbalance between a user and the whole network. The max-

imum service imbalance allowed by NABT is determined by the underlying social network topology and different credit limits on social links. In fact, for a trading system with a centralized bank and a global currency, service sinks can “buy” services as much as their balances with the bank allow. One can treat such a system as a trading network with a star topology rooted at the bank. The trading efficiency of the bank system is also limited by the min-cut of the star topology. As will be shown in our simulations, the efficiency of NABT in well-connected online social networks is very close to trading systems with banks and global currencies.

#### F. Dynamic Credit Routing

To study the efficiency of NABT, we assumed a static service demand and the credit transfer routing is calculated using a centralized algorithm to maximally satisfy service demands. In reality, users generate service demands dynamically. Under dynamic service demands, credit transfers between users can no longer be grouped into a single time slot and executed simultaneously. Instead, credit transfer routing has to be done dynamically. Credit routing for a new service demand takes as given the credit balances on all links after credit transfers for earlier demands. If no legitimate credit flow can be found for a peer’s demand, it can be either dropped, or kept in the peer’s request buffer to wait for credit balance changes triggered by future transactions from other peers in the network. To study the efficiency of NABT with dynamic demands, we employ a time-slotted model. At each time slot  $k$ , a new set of demands  $H(k)$  is generated among users. Given the credit balance on all social links resulting from previous services, our goal is now to find credit transfer flows to maximally satisfy current demand set. We show the memoryless property of credit routing.

*Theorem 3:* Given a set of executed demands in history  $H(i)$ ,  $1 \leq i \leq k-1$ , a new demand set  $H(k)$  is executable in  $G_S$  if and only if the aggregate demand up to time  $k$ ,  $A(k) = \sum_{i=1}^k H(i)$ , is executable in  $G_S$  with zero initial credit balance on all links. In other words, the executability of  $H(k)$  is independent of how credit flows were set up for demand sets  $H(i)$ ,  $1 \leq i \leq k-1$ , that have been executed in the past.

*Proof:* Denote by  $\mathcal{X}(i)$  the credit flow for demand set  $H(i)$ ,  $1 \leq i \leq k-1$ .

**ONLY IF  $\Rightarrow$ :** If  $H(k)$  is executable at time  $k$ , let  $\mathcal{X}(k)$  be the corresponding credit flow. Naturally we have a credit flow for the aggregate demand  $A(k)$  in the initial social graph by merging  $\{\mathcal{X}(i), 1 \leq i \leq k\}$  into one set of credit flows.

**IF  $\Leftarrow$ :** Let  $\mathcal{Y}(k)$  be the credit flow for the aggregate demand set  $A(k)$  in  $G_S$  with zero credit balance on all links. Similar to the previous argument, all credit flows configured up to time  $k-1$  execute the aggregate demand up to time  $k-1$ :  $A(k-1) = \sum_{i=1}^{k-1} H(i)$ . Let  $\mathcal{Y}(k-1)$  be the aggregate credit flow. Construct a new demand  $\check{A}(k-1)$  by reversing the directions of all demands of  $A(k-1)$ , construct a new flow  $\check{\mathcal{Y}}(k-1)$  by reversing the directions of all credit flows in  $\mathcal{Y}(k-1)$ , immediately  $\check{\mathcal{Y}}(k-1)$  implements all demands in  $\check{A}(k-1)$ . At time  $k$ , we first add in credit flow  $\check{\mathcal{Y}}(k-1)$  to execute

$\check{A}(k-1)$ , then all links’ credit balance go back to zero. Then we add in  $\mathcal{Y}(k)$  to implement  $A(k)$ . In two steps, we construct a credit flow for the composite demand  $\check{A}(k-1) + A(k)$ , which has the same reduced demand matrix as  $H(k)$ . According to Lemma 3,  $H(k)$  is also executable at time  $k$ . ■

With the memoryless property, the credit routing feasibility for new service demands is independent of how credit routing was done for demands in the past. We have therefore shown that, at any given time  $k$ , all feasible routing solutions for the current demand set  $H(k)$  are equally good for future routing.

#### IV. TRACE-DRIVEN SIMULATION

Having established some of the mathematical properties of NABT, we now explore more closely its potential use within an online social network. To this end, we use P2P file sharing as an example to demonstrate the applicability and efficiency of NABT. We simulate a P2P file sharing network with 10,000 peers. Each peer dynamically generates requests to download MP3 music files in a catalog of 10,000 files. We investigate and compare three trading paradigms:

- **Trading with global currency (GCT).** We assume there is global currency. Peer  $i$  has  $B_i$  initial credits (monetary units). For simplicity, we assume that each MP3 file costs one credit. Any peer can download a file from any other peer. If peer  $i$  wants to download a file from peer  $j$ ,  $i$  pays one credit directly to  $j$ .
- **Synchronous trading (ST).** Two peers can trade if and only if they can supply files to each other simultaneously. Each peer can trade with all other peers in the system.
- **Two-hop NABT.** Peers are connected in an underlying social network. For each file download request, a requesting peer inquires with its friends (called one-hop friends) and the friends of its friends (called two-hop friends) in the social network. The peer checks whether there are potential supplying peers within the two hops, and whether the paths to these supplying peers can pass sufficient pairwise credits. If there exists multiple paths, the requesting peer randomly selects a path.

#### A. Handling Disputes: Dynamic Credit Limit Setting

In NABT, the credit limits  $C_{ij}$  are individually set by the users and reflect the willingness of trading between friends. With dynamic service demands, users dynamically negotiate credit limits with their friends. On one hand, similar to practices in real social networks, a pair of friends are more willing to trade with each other as more trades between the two, direct or indirect, have been fulfilled. On the other hand, although it is anticipated that most of users in online social networks will cooperate with their friends, it is inevitable that there exists uncooperative users, such as free-riders. It is desirable of NABT to punish these uncooperative users. To this end, we adopt a simple *Additive Increase Multiplication Decrease (AIMD)* adjustment algorithm: the credit limit on a social link increases by an amount of  $\alpha$  after each fulfilled transaction utilizing the link; the credit limit decreases by a factor of  $\beta$  whenever the link is involved in an unfulfilled

or disputed transaction. This way, users are further motivated to fulfill services requested from them and relay credits for their friends. Cheating peers will be punished by losing their connectivity for future service trades. To avoid credit limit explosion, a maximum credit limit can also be set.

### B. Simulation Setup

Time is advanced in time slots, with each time slot representing one minute. We simulate the system for 2,880 minutes in each simulation run. To simplify the simulation and focus on the P2P trading efficiency, we assume files have the same size of 3 MB (about the size of a typical MP3). In our simulations, we assume the file popularity follows a Zipf distribution, with the parameter  $\rho = 0.27$  [4]. There are two types of peers in terms of bandwidth: 37% of peers are Ethernet users with an upload bandwidth contribution of 1.2 Mbps, while the rest of peers are residential users with an upload bandwidth contribution of 400 kbps [12]. In terms of willingness for sharing, we also assume that there are two types of peers: 10% of the peers are *content-rich peers*, each sharing 1,000 files; 90% of the peers are *content-scarce peer*, each sharing 50 files. Initially, each peer is assigned a random subset of files for sharing based on the Zipf distribution. The cached files at each peer evolve during the simulation. The oldest files in the cache are replaced by the newly received files. In the system, each peer goes online and offline randomly, following a Markov On-OFF process with both the average on time  $T_{on}$  and the average off time  $T_{off}$  set to 720 minutes. When a peer re-enters the system, it has the same cached files and credit balances as when it last left the system.

Each peer generates file download requests according to a Poisson process with rate  $\gamma$ . By default, a peer requests a new file every 10 minutes on average. After a peer generates a new file request, it attempts to find a supplier who can serve this request. A peer may need to request this file multiple times in the future, since it may not be able to find a suitable supplier immediately. However, if a request cannot be scheduled to a supplier within 60 minutes after it has been generated, the peer will simply *drop* this request and will not request it anymore. We assume that each peer can at most manage 100 unscheduled requests. If the number of unscheduled requests reaches 100, any newly generated requests will be simply *blocked*.

Meanwhile, a supplier may have multiple requests from multiple peers. Similar to most of the P2P file downloading applications, the supplier maintains a *serve queue* to manage the received requests from other peers. Requests in the serve queue are served in a first-in-first-out (FIFO) fashion. For a particular request, when there is more than one supplier that satisfies a given trading policy, the requesting peer will select a supplier that can serve this file with the shortest waiting time.

To simulate NABT, we assume peers are connected by an underlying online social network with a topology collected from MySpace. The trace is obtained by crawling the MySpace on-line Web site from September to October, 2006 [1]. The crawler randomly selects a starting user, crawls the user's

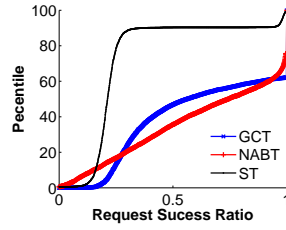


Fig. 5. CDF of request success ratio for different systems.

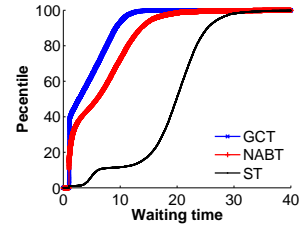


Fig. 6. CDF of waiting time for different systems.

friends' pages, their friends' pages, and so on. In the MySpace trace, a user has 127 friends on average. However, a large fraction of the users have a small number of friends; for example, more than 50% of users have less than 10 friends. In each time slot, we only consider friends who are currently online. The default initial credit limit  $C$  for each pair of peers is set to 5. After each transaction, all peers on the path adapt their credit limits according to the AIMD algorithm described in Section IV-A, with  $\alpha = 0.1$  and  $\beta = 0.5$ . The maximum credit limit is set to 100. To make the comparison as fair as possible, each peer  $i$  in GCT is given an amount of initial credits equal to the sum of all of peer  $i$ 's initial pairwise credits in NABT with all of its active friends. Thus,  $B_i$  is set to be  $CD_i T_{on} / (T_{on} + T_{off})$ , where  $D_i$  is the number of friends of peer  $i$  in NABT.

### C. Simulation Results

1) *Trading Efficiency Comparison:* We compare the trading efficiency of the three paradigms mainly using two performance metrics:

**Request success ratio ( $\theta$ ):** The ratio of fulfilled requests to the total number of requests for a peer. Note that a request cannot be fulfilled in our file sharing systems in two cases: (i) the request cannot be scheduled within 60 minutes after it has been generated; (ii) after the request is generated, there has already been 100 pending requests.

**Average waiting time ( $T$ ):** The waiting time for a fulfilled file download request is the time lag between the request arrival and the download completion time. Since the files are small, the actual file transmission time is short. The waiting time is mostly due to (i) the time that the peer spends on finding a suitable supplier and (ii) the delay in the serve queue in the supplying peer.

Figure 5 and 6 show the CDF of  $\theta$  and  $T$  across all peers for the NABT, ST, and GCT schemes. From Figure 5, we observe that the ST system generally has a low request success ratio. As expected, the lack of common interest in the ST system limits the chance of trading, especially for the content-scarce peers. The content-rich peers have a much higher probability for trading and consequently have a high  $\theta$ . From Figure 6, we can see that the ST system has a long waiting time  $T$  given that a request can be successfully scheduled. We should note that our simulated system is underloaded (the system utilization is about 0.05), so that long  $T$  is mainly contributed by the

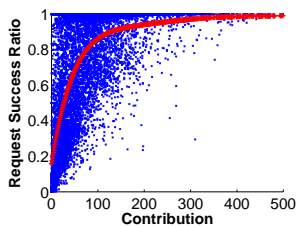


Fig. 7. Relation between request success ratio and upload contribution

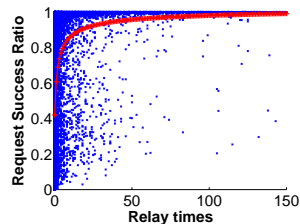


Fig. 8. Relation between request success ratio and relay contribution

waiting time at a requesting peer before this request has been successfully scheduled, instead of the waiting time in the serve queue at a supplying peer. This is because a request peer has to wait for a long time to find a trading partner with mutual interest. The low trading efficiency of ST results in low request success ratio and long waiting time.

The GCT system has a much higher request success ratio, as indicated in Figure 5. These peers have enough credits to pay for all of the files they want. They either have a large amount of initial credits, or earn sufficient amount of credits during the trading process with a high serving capacity. The peers with less credits or lower serving capacity cannot support all of their requests, resulting in a relatively low  $\theta$ . Figure 6 shows that peers in GCT also experience short waiting times  $T$ . After a peer locates a supplying peer, as long as it has enough credits in the bank, it can schedule the requested file immediately.

Figure 5 and 6 show that the efficiency of NABT is much better than ST and is very close to GCT. In NABT, a peer with enough pair-wise credits has a very good chance to pass the credits to the supplying peer. For peers not well connected in the social network, they need to spend more time before their file requests have been scheduled on obtaining an available path to pass the credits. As a result, the waiting time of NABT is slightly longer than GCT.

2) *Service Differentiation*: A good incentive design leads to service differentiation among peers with different service contributions. Figure 5 and 6 show that different peers have different success ratios and waiting times. In this section, we will see whether these different level of service quality relate to the peers' contributions to the system. Figure 7 relates the success ratio of each peer to its upload contribution (in terms of number of files). The red curve is obtained by using power fitting on all of the scattered points. Peers with a higher contribution are more likely to have a higher request success ratio. This is especially true for the peers that contribute more than 300 files: they successfully schedule almost all of their requested files.

Figure 8 relates the request success ratio of each peer to its relay contribution (in terms of the number of times of being an intermediary peer). Similar observations can be made: peers that are willing to help their friends for relaying the pair-wise credits are more likely to have a higher success ratio. Our

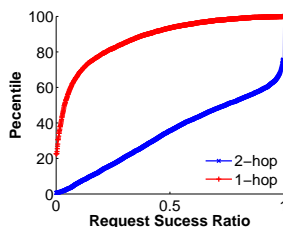


Fig. 9. CDF of request success ratio with and without intermediaries.

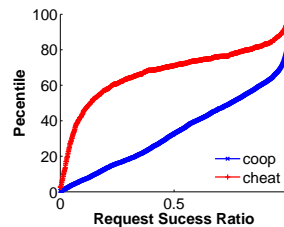


Fig. 10. CDF of request success ratio with the presence of free-riders.

AIMD credit limit adaptation algorithm increases the credit limits for the trading intermediaries. This provides incentives for the peers to act as trading intermediaries.

3) *Importance of Trading Intermediaries*: Intermediaries play important roles in NABT. In this section, we examine the efficiency improvement brought in by intermediaries. We conduct simulations to compare the performance of two-hop NABT with direct bilateral trading. In direct bilateral trading, a peer is only permitted to trade with its one-hop friends. Figure 9 shows the CDF of  $\theta$  across all peers for the direct bilateral trading and the two-hop NABT systems. We can observe that the performance of direct trading is much worse than two-hop NABT. Intermediaries improve the trading efficiency of direct bilateral trading in two ways: (i) Networked trading significantly increases the trading coverage. The ratio between the number of two-hop neighbors and the number of one-hop neighbors is approximately the average number of active friend links of a peer in the social network. In our simulations, the average number of friend links is 127, and about half of them are active at any given time. Therefore one intermediary can increase the number of potential trading partners for a peer by a factor of more than 60. (ii) Intermediaries also increase the number of credits that can be passed between direct neighbors. It is possible for a peer to pass credits to its one-hop friends using multiple parallel two-hop paths through intermediaries. This effectively increases the aggregate credit limits between a pair of direct friends.

4) *Impact of Cheating*: We now consider free-riding and cheating behavior. In a P2P file sharing system, a free-rider may try to receive the same or even better services as regular peers with minimum contribution to the system. Although a social network provides built-in reputation among friends, it nevertheless still needs some mechanism to limit uncooperative behaviors. For example, as a supplying peer, a free-rider may not fulfill its commitment to serve a file after it receives the credits; as a requesting peer, a free-rider may not pay the credits after it receives the requested file; as an intermediary peer, a free-rider may not pass the pair-wise credits properly. These uncooperative behaviors will lead to a *dispute* among the peers on the credit transfer path. We now show that NABT adapts the credit limits to limit the obtainable service quality of cheaters.

In our simulation, we assume that 10% of the peers are free-riders. A free-rider behaves as a cheater, causing a dispute,

with probability of 0.5. We investigate whether the system can provide a limited service quality to the free-riders. Figure 10 plots the CDF of request success ratio for the cooperative peers and the free-riders. We can observe that with our AIMD credit limit adaptation algorithm, (i) the free-riders receive much worse service compared with the cooperative peers; and (ii) the cooperative peers have a similar  $\theta$  as when there are no free-riders (Figure 5). Since a free-rider keeps cheating, most of its friends block its file or credit transfer requests by reducing its credit limit. Although the cooperative peers may also be punished when getting involved in a dispute, the disputes nevertheless have little affect on their overall trading efficiency. This is because: (i) for a cooperative peer, the occasional credit limit reduction of a particular friend does not affect its credit limits with its other friends; (ii) the cooperative peer can rectify its reduced credit limit by serving file requests and/or transferring credits.

## V. RELATED WORK

There are several studies on BitTorrent's tit-for-tat incentive mechanism [6], [14], [7], [15], [16], [19], [20]. As described in Section 1, BitTorrent's tit-for-tat scheme is essentially synchronous bartering; as demonstrated in our simulations, synchronous schemes are generally inefficient. There have been several important studies of asynchronous incentive schemes in P2P systems. For example, there are several proposals for using global currency in P2P [23], [22], [2], [10]. NABT, with its distributed currency and bilateral trading, is very different from all of these global currency schemes. There has also been interesting proposals for global P2P reputation systems [13], [11]; but to date there isn't a large-scale P2P deployment that has successfully used reputation. Game theoretic approaches for P2P incentives have been studied in [17], [8], [9]. Feldman et al use simulation to show that reciprocation incentive mechanisms can drive the system of strategic users to nearly optimal levels of cooperation [9]. More recently, Zhao et al developed a mathematically-tractable dynamic game-theoretic framework to analyze a broad class of P2P incentive schemes [25]. Built-in trustworthiness in online social networks has been widely studied to enhance security in computer networks [3], [21], [24]. To our knowledge, this paper, proposing and exploring NABT, is the first to examine P2P incentives in the context of online social networks.

## VI. CONCLUSIONS

We presented a new P2P trading paradigm for online social networks: Networked Asynchronous Bilateral Trading (NABT). We showed that NABT solves two fundamental problems of traditional synchronous bilateral P2P trading: asynchronicity over time and asynchronicity over nodes. Supporting a form of "network tit-for-tat" we mathematically showed that NABT is perfectly efficient with balanced demands. We then showed that the efficiency of NABT with unbalanced demands is determined by the min-cut between service sources and service sinks. Through simulations driven by a MySpace trace, we showed that NABT greatly improves

the efficiency of synchronous bilateral trading and can achieve almost the same level of efficiency as global currency. In addition, NABT can effectively isolate cheaters and motivate peers to cooperate in service trading and credit transfer. We are currently implementing NABT algorithms as incentive engines for P2P applications in on-line social networks. The implementation of NABT is straightforward and only requires minimum coordination among friends in social networks.

## REFERENCES

- [1] Y.-Y. Ahn, S. Han, H. Kwak, S. Moon, and H. Jeong, "Analysis of topological characteristics of huge online social networking services," in *WWW*, 2007.
- [2] C. Aperia, M. J. Freedman, and R. Johari, "Peer-assisted content distribution with prices," in *Proceedings of CoNext*, 2008.
- [3] L. Banks, P. Bhattacharyya, M. Spear, and S. F. Wu, "Davis social links: Leveraging social networks for future internet communication," in *Ninth Annual International Symposium on Applications and the Internet*, 2009.
- [4] J. Chu, K. Labonte, and B. N. Levine, "Availability and locality measurements of peer-to-peer file systems," in *ITCom*, 2002.
- [5] T. Clark, "Digicash files chapter 11," <http://news.cnet.com/2100-1001-217527.html>.
- [6] B. Cohen, "Incentives build robustness in BitTorrent," in *Workshop on Economics of Peer-to-Peer Systems*, June 2003.
- [7] B. Fan, D.-M. Chiu, and J. C. S. Lui, "The delicate tradeoffs in BitTorrent-like file sharing protocol design," in *IEEE ICNP*, November 2006.
- [8] M. Feldman, K. Lai, I. Stoica, and J. Chuang, "Robust incentive techniques for peer-to-peer networks," in *EC*, 2004.
- [9] M. Feldman, C. Papadimitriou, J. Chuang, and I. Stoica, "Free-riding and whitewashing in peer-to-peer systems," in *Workshop on Practice and theory of incentives in networked systems*, 2004.
- [10] P. Golle, K. Leyton-Brown, and I. Mironov, "Incentives for sharing in peer-to-peer networks," in *ACM Conference on Electronic Commerce 2001*, 2001.
- [11] M. Gupta, P. Judge, and M. Ammar, "A reputation system for peer-to-peer networks," in *Proceedings of NOSSDAV*, 2003.
- [12] C. Huang, J. Li, and K. W. Ross, "Can Internet VoD be profitable?" in *ACM SIGCOMM*, Kyoto, August 2007.
- [13] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The eigentrust algorithm for reputation management in P2P networks," in *Proceedings of WWW*, 2003.
- [14] A. Legout, N. Liogkas, E. Kohler, and L. Zhang, "Clustering and sharing incentives in BitTorrent systems," in *ACM SIGMETRICS*, June 2007.
- [15] N. Liogkas, R. Nelson, E. Kohler, and L. Zhang, "Exploiting BitTorrent For Fun (But Not Profit)," in *Proc. of IPTPS*, 2006.
- [16] T. Locher, P. Moor, S. Schmid, and R. Wattenhofer, "Free riding in BitTorrent is cheap," in *ACM HotNets 2006*, Irvine, November 2006.
- [17] T. B. Ma, C. M. Lee, J. C. S. Lui, and K. Y. Yau, "Incentive and service differentiation in P2P networks: A game theoretic approach," *IEEE/ACM Trans. on Networking*, vol. 14, no. 5, October 2006.
- [18] A. Mas-Colell, M. D. Whinston, and J. R. Green, *Microeconomic Theory*. Oxford University Press, 1995.
- [19] M. Piatek, T. Isdal, T. Anderson, A. Krishnamurthy, and A. Venkataramani, "Do incentives build robustness in bittorrent," in *Proc. of NSDI*, 2007.
- [20] M. Sirivianos, J. H. Park, R. Chen, and X. Yang, "Free-riding in BitTorrent networks with the large view exploit," in *IEEE IPTPS*, February 2007.
- [21] D. N. Tran, B. Min, J. Li, and L. Subramanian, "Sybil-resilient online content rating," in *NSDI*, 2009.
- [22] V. Vishnumurthy, S. Chandrakumar, and E. G. Sirer, "Karma: A secure economic framework for peer-to-peer resource sharing," in *Proceedings of P2P-Econ*, 2003.
- [23] B. Yang and H. Garcia-Molina, "PPay: Micropayments for peer-to-peer systems," in *Proceedings of CCS*, 2003.
- [24] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman, "SybilGuard: Defending against sybil attacks via social networks," in *SIGCOMM*, 2006.
- [25] B. Q. Zhao, J. C. S. Lui, and D.-M. Chiu, "Mathematical modeling of incentive policies in P2P systems," in *NetEcon*, 2008.