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

Zhengye Liu[†], Hao Hu[†], Yong Liu[†], Markus Mobius[*], Keith Ross[‡] and Yao Wang[†]
[†]Electrical & Computer Engineering, Polytechnic Institute of NYU
[*]Department of Economics, Harvard University
[‡]Computer Science and Engineering, Polytechnic Institute of NYU

## ABSTRACT

The success of future P2P applications ultimately depends on convincing users to volunteer their bandwidth, CPU and storage resources, which is the challenge of incentive design. A natural approach to providing incentives in P2P applications is to use a single *global currency*, whereby peers earn currency units when contributing resources. Global currencies, however, require a high degree of coordination, hindering their deployment in large-scale and decentralized P2P systems. At the other extreme of global currency is *barter*, for which peers bilaterally and synchronously trade resources. Although bartering is simple and scales, it does not fully incentivize peer contribution. In this paper, we propose a new incentive paradigm, *Networked Asynchronous Bilateral Trading (NABT)*, that can be applied to a broad range of P2P applications. NABT is applicable in online social networks. In NABT, each pair of friends keeps track of a credit balance between them. When user Alice provides a service (a file, storage space, computation, expert advice and so on) to her friend Bob, she charges Bob credits. Unlike synchronous barter, NABT allows peers to supply each other at different points in time. NABT further allows peers to trade with remote peers through intermediaries. We theoretically show that NABT can achieve the same level of trading efficiency as global currency systems. Using simulations driven by MySpace traces, we demonstrate that a simple two-hop NABT design can greatly improve the efficiency of synchronous bilateral trading and effectively punish cheaters. The proposed NABT framework can be easily adopted by contemporary online social networks to provide incentives for existing and new P2P applications.

## 1. 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. Ultimately, P2P can potentially realize the Worldwide Computer [3], a transformative vision in which billions of peer components – collectively made available to applications through a common API – to provide an infrastructure for file distribution and storage, live and on-demand video streaming, VoIP, distributed computation, and so on.

But the success of future P2P applications ultimately depends on convincing users to volunteer their resources, 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. With global currency, Alice, as a resource supplier, would earn currency units when she transmits bits, stores bytes, provides computation and so on for the benefit of P2P application. She can then, in turn, use her earned currency to receive bits, have files stored, or have computation performed by the P2P application. Such 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 [7]. From economic theory, we know that global currency systems are highly efficient [27].

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 of currency which would ignite inflation and undermine the usage of currency. It is also difficult to justify the high cost of maintaining a complex currency and banking infrastructure for P2P applications that normally trade goods and services in high volume but of small value. Although there have been several proposals for using global currency in P2P [35, 34, 4], and several systems have actually been built, there hasn't been a deployment to date that has taken hold on a large scale [28, 7]. There has also been numerous proposals for global P2P reputation systems [18, 16]; but to date there isn't 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 cur-

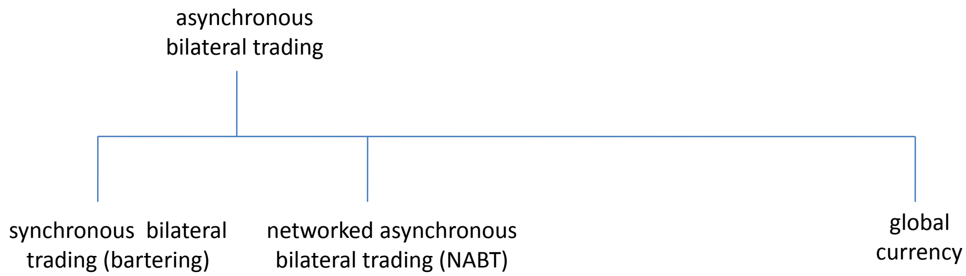



**Figure 1: The spectrum of economic systems.**

rency, banks, public-key infrastructures, elaborate dispute-resolution mechanisms or bank regulation. To date, there is one large-scale P2P deployment with a "partially" successful bartering system, namely, BitTorrent. Fundamentally, BitTorrent's incentive principle is as follows: a peer will get the file faster if it contributes more upload bandwidth to the torrent. This incentive encourages users to add resources to the torrents they participate in, for example, to upgrade their ISP access or increase the maximum upload rates in their BitTorrent clients. BitTorrent and its variants realize this basic incentive using a form of bilateral trading, sometimes referred to as tit-for-tat, for which Alice gives more chunks to those peers who are currently giving her the most chunks [8, 23, 10]. This trading is *synchronous* because it is taking place within very short time periods, on the order of tens of seconds. BitTorrent is wildly successful, with millions of simultaneous peers actively sharing files in hundreds of thousands of swarms. Clearly, if BitTorrent had been designed without tit-for-tat (or a variant), but otherwise exactly the same, BitTorrent would have been a major failure, as the majority of the users would have been free-riders [1, 32].

BitTorrent's incentive scheme is only "partially" successful in that its bilateral incentive scheme can be circumvented [24, 25, 31, 33]. Moreover, it is difficult to trade long-tail content with BitTorrent because users demands' are rarely bilaterally matched and there is no incentive for a peer to contribute if it has no immediate demand. In addition, BitTorrent's synchronous bilateral trading is expressly designed for file sharing and is not appropriate for a wider range of applications. For example, tit-for-tat is inappropriate for live streaming applications, in which high-bandwidth uploaders would have no incentive to supply bandwidth beyond what is needed to receive the video at the full rate; and low-bandwidth uploaders might not even be able to trade at a sufficiently high rate to see the video. More generally, synchronous bilateral trading does not facilitate a market in which users trade diverse and mixed resources (bandwidth, content, computation, storage, expertise, and so on).

As illustrated in Figure 1, in the spectrum of economic systems, 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)*, that expands synchronous bilateral trading by combining it with *asynchronicity*, *network trading via intermediaries*, and *online social networks*. In NABT, pairs of friends in online social networks maintain a credit balance with each other, allowing them to supply each other at different points in time.

Bilateral credit alone, however, does not completely solve the market efficiency problem, because any specific pair of friends can only trade rarely with each other over time. NABT increases the market efficiency by *allowing trade 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 friend Charlie. As discussed in Section 2, NABT abounds in real-world economies, and can serve as inspiration for P2P incentive design. NABT is particularly appropriate in the context of online social networks (such as Facebook and MySpace), where users have a natural set of friends with whom they can set debt limits.

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 synchronous bilateral P2P trading: asynchronicity over time and asychronicity over nodes.

- We develop formal models to study the efficiency 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 discuss various practical NABT design considerations in dynamic P2P trading environment. We establish the memoryless property of dynamic credit transfer routing in NABT. Distributed dynamic credit routing and credit limit adjustment algorithms are proposed.

- We apply two-hop NABT to P2P file sharing applications. Through extensive simulations driven by a large scale MySpace trace, we show that NABT greatly improves the efficiency of synchronous bilateral trading and can achieve almost the same level of efficiency as global currencies. NABT can effectively isolate cheaters and motivate peers to cooperate in service trading and credit transfer.

The rest of the paper is organized as follows. In Section 2, we introduce NABT as a new P2P incentive paradigm in online social networks. We describe the fundamental incentive underpinnings of NABT in Section 3. The efficiency of NABT under balanced and unbalanced demand sets are analyzed in Section 4. Practical NABT design considerations are investigated in Section 5. The performance of two-hop NABT in P2P file sharing is evaluated through simulations driven by the MySpace trace in Section 6. We summarize the related work on P2P incentive in Section 7. The paper is concluded in Section 8.

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

In the context of P2P sharing, synchronous trading is appropriate when peers are synchronized in terms of both time and service interest (content, computation, storage, etc.) But in most other P2P resource markets that involve long-tail content and diverse service interest, synchronicity is highly constraining and inefficient. Synchronous trading fails along two dimensions: (1) demand requests occur at different times (asynchronicity across time) and (2) users' demand and supply are not typically bilaterally matched over time (asynchronicity across nodes or absence of "double coincidence of wants" in economic terms).

Real economies have developed an an intriguing class of alternative trading systems in situations where there is no central authority to support a global currency that are colloquially referred to as **trading favors** [29, 30, 21]. They are common in developing countries and within professional networks. People in developing countries often lack liquidity which makes it difficult to participate in formal markets and requires them to procure services through their direct and indirect social network [9, 22, 20].[1] For example, Alice might want to borrow a truck but none of her direct friends has a truck. But she recently helped out her friend Bob who has a friend, Charlie, with a truck. Bob can act as an intermediary, call in a favor from Charlie and trade a favor with Alice. For professional networks, it was shown that lawyers "trade" clients to efficiently pool similar claims for class-action lawsuits [13]. Favor trading is also commonly used when looking for a job [15, 5].

Intuitively, favors solve the problem of asynchronicity across time while the use of intermediaries solves the long-tail problem or asynchronicity across nodes.

### 2.1 NABT Framework

In this section we describe the basic elements of *NABT* which allow us to make the the favor trading paradigm of real-world social networks operational for online social networks. Facebook, MySpace and other online communities present exciting new opportunities to adopt NABT for P2P incentive design.

1. *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.

2. *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[2].

3. *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
$$-C_{ji} \leq b_{ij} \leq C_{ij}. \quad (1)$$

4. *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 $C_{ij}$.

5. *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[3]. 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

[1] Often, formal markets for certain services such as renting cars, trucks or tools do not even exist in developing countries because of poorly functioning legal systems.

[2] 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.

[3] If Alice is providing Charlie a file, the transfer would normally be done directly, from peer to peer, without passing through Bob. But the credit transfer would take place via Bob.

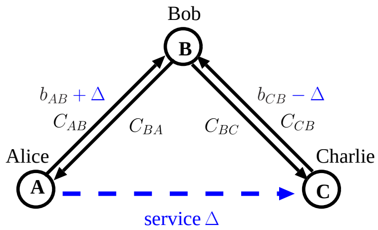



**Figure 2: Example of Credit Transfer in Social Network**

asynchronous trading, and trading of mixed services. Analogous to "favor trading" in real economies, it is straightforward to implement within the context of an online social network. But before NABT can become a viable solution, a number of fundamental questions need to be addressed. First, does NABT's credit limit structure provide true incentives to users for sharing goods and services? How resilient is NABT against cheating of individual peers? Second, under what conditions does NABT market efficiency approach that of global currency? Third, what is a simple and efficient algorithms to enable networked trading? In the following sections, we will answer these questions using analysis and simulations.

# 3. DOES NABT INCENTIVIZE?

In this section we address the fundamental question of whether NABT provides robust incentives for engineering a flourishing economy. To examine this question, we discuss some of the fundamental underpinnings of asynchronous trading as they apply to both off-line and on-line social networks.

## 3.1 Asynchronous Trading

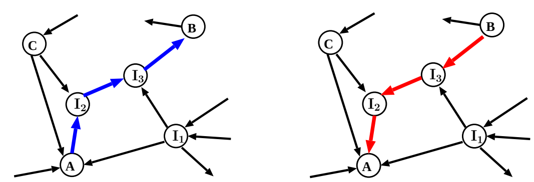
When trading is asynchronous over time but not across nodes the theory of repeated games provides a simple bilateral incentive-compatible mechanism that naturally gives rise to credits. Consider the example of two users, $A$ and $B$, who each have a list of files. Both users have demands for files that arise independently (and therefore asynchronously across time) at constant rate 1. Users cannot verify which files are available for download from the other user but they know that each requested file is available with probability $p$. Downloading a file has utility $u$ and sending a file has cost $h < u$. An incentive compatible mechanism has to satisfy two criteria. (1) The mechanism has to provide incentives to disclose that a requested file can be in fact supplied, and (2) the potential supplier must be willing to bear the cost $h$ of uploading the file. The parameter $p$ codes the degree to which users' needs are mutually compatible: when $p$ is large both users can satisfy most of their file-sharing needs through mutual trade and asynchronicity across nodes is low. A simple credit mechanism provides proper incentives to both users: users initially start with a credit score of 0 and count the net number of "favors" (or downloads) provided by both partners. As long as the credit balance stays strictly within the interval $[-C, C]$ for some integer $C > 0$ agents voluntarily disclose whether they possess a requested file and incur the upload cost. If the credit balance hits the upper or lower limit the agent who provided a surplus of uploads in the past stops further uploads until the other agent built up some credit for proof.

THEOREM 1. *Let $r$ be the user future utility discount rate. A Markov-perfect equilibrium with voluntary disclosure and uploads exists if $C < \overline{C}(\frac{r}{p}, \frac{u}{h})$ where the upper bound $\overline{C}$ is decreasing in its argument $\frac{r}{p}$ and increasing in its argument $\frac{u}{h}$. (See [29] for proof.)*

This result illustrates a basic trade-off between efficient trading and providing incentives: large credit limits make credit limits less constraining but they also reduce agents' incentives to disclose and upload files.

## 3.2 Network Trading

Bilateral trading fails when there is asynchronicity across both time and nodes such that $p$ is small [19]. Demand patterns for most long-tailed content exhibit a small $p$, since users' tastes for content are highly diverse. Consider the ex-




(a) before transaction (b) after transaction

**Figure 3: Example of Trading Favors in Social Networks**

ample of user A who is searching for a file that happens to be on user B's hard drive. In this example, the probability $p$ with which A can find some file of interest on B's hard drive is small. According to Theorem 1, this makes it difficult to build a bilateral long-term trading relationship with that agent: neither A nor B would have an incentive to disclose files on their hard drive and incur the cost of uploading them. However, every user now has four fixed friends with whom he shares a credit balance. For the sake of exposition we choose a very simplified credit model: an agent either owes a favor (incoming arrow) or is owed a favor (outgoing arrow). We say that agents A and B are connected through an open path if there is a directed chain of outgoing links connecting A and B. In our example, the chain $A \rightarrow I_2 \rightarrow I_3 \rightarrow B$ in Figure 3(a) satisfies this criterion. The existence of this chain can be revealed, for example, if A starts a search for the MP3 file and his friends delegate the request to their friends and so on, until the relayed request reaches B. If B discloses possession of the file and allows A to download it we propose the following accounting protocol: all links along the open chain connecting A and B change direction, as illustrated in Figure 3(b). Intuitively, A has used up one of his favors and B has gained a favor. The two intermediaries $I_2$ and $I_3$ have each traded one favor for the other. Authors of [30] derived the bounds on the probability that a user is connected to another user who possesses a requested file. Importantly, the high chance of being able to find a resource through an open link provides incentives to trade favors with neighbors. In particular, user B has an incentive to voluntarily disclose possession of the file and send it to $A$ because he gains something valuable in return: an open link to his friend $I_3$ which he can use for future trading.

It has been shown in [29, 30] that favor trading has a number of interesting properties: (i) Favor trading creates stable long-term trading relationship with friends. (ii) Favor trading can be thought of as creating personalized "monies" between pairs of agents. Relaying favors is akin to trading one of personal money for another unit. In this sense favor trading implements "favors" as a type of global currency which does not require a central authority. (iii) The mechanism is coalition-proof in the sense that groups of agents cannot jointly deviate by redefining the direction of links between themselves. Such operations do not change the net wealth of the group measured by the number of net favors with the rest of the community. (iv) Weak-link network topologies (such as random networks) provide better incentives because the overlap between the set of agents that can be reached through two separate open links is small and the marginal value of each link is therefore large [15].

## 4. EFFICIENCY OF NABT

One immediate question for online NABT is how efficient it is compared with global currencies. The efficiency of NABT is determined by three major factors: $(i)$ *Network geometry:* How are peers connected in online social network? $(ii)$ *Credit limits:* How high are credit limits among pairs of friends? $(iii)$ *Demand and supply distribution:* How are service demands and supplies distributed? Given these constraints, credit flows should be arranged for all service demands to maximize the economic activity. In this section, we analytically study the *efficiency*. To this end we need to first introduce a formal model for Online NABT.

### 4.1 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. If user $i$ treats $j$ as his friend, $i$ connects to $j$ through a directed link $\langle i, j\rangle \in F$. Since friendship is mostly a mutual relationship, there exist a pair of directed links between two friends. Suppose each user can provide some services, and also want 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 5.1 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 $(k, l) \notin D$.

Assume there is no centralized bank and no common currency in the system. User $k$ and $l$ can trade their services synchronously if they need services from each other and $h^{(k,l)} = h^{(l,k)}$. However, this synchronous trading is too restrictive. With the pairwise credits introduced in Section 2.1, 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, $p(k, l) = \{k = r_0 \to r_1 \to r_2 \to \cdots r_{m-1} \to r_m = l\}$. Since friend links are bi-directional, $l$ can initiate a sequence of credit transfers in the reverse direction of path $p(k, l)$: $r_n$ borrows $h^{(k,l)}$ credits from $r_{n-1}$, $n = m, \cdots, 1$. After the credit transfers, $k$ can provide the service to $l$ and the credit balance on each node is updated as $b_{r_n r_{n+1}}(t + \delta) = b_{r_n r_{n+1}}(t) + h^{(k,l)}$.[4]

### 4.2 Credit Transfer Routing in NABT

More generally, a service demand 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 $x^d_{\langle i,j\rangle}$, which is defined as the amount of credits that node $j$ borrows from node $i$ for service $d$. $\{x^d_{\langle i,j\rangle}, \langle i, j\rangle \in F\}$ should satisfy the *flow conservation* on all nodes in the network:

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

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 $\sum_{d\in D} x^d_{\langle i,j\rangle}$ and $\sum_{d\in D} x^d_{\langle j,i\rangle}$ respectively.

[4]Pairwise credits between different friend pairs are not exchangeable. Therefore, the credit balances between different friend pairs cannot be merged.

Therefore the resulted credit balance between user $i$ and $j$ after all credit transfers is $b_{ij} = \sum_{d\in D}(x^d_{\langle i,j\rangle} - x^d_{\langle j,i\rangle})$. We have to make sure that the resulted credit balances on all links after executing all services are bounded by their credit limits:

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

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 $\mathcal{X}(H) \triangleq \{x^d_{\langle i,j\rangle}, d \in D, \langle i,j\rangle \in F\}$ 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.

## 4.3 Efficiency with Balanced Demand Set

In an ideal synchronous bilateral trade, a peer plays tit-for-tat with another peer and the pairwise service contribution and consumption balance out. In this section, we show that NABT supports **networked tit-for-tat**: *NABT is perfectly efficient if the aggregate service consumed by each user (regardless of the supplier) equals to the aggregate service supplied by him (regardless of the consumer).*

DEFINITION 1. *Balanced Demand Matrix: a demand matrix $H$ is called balanced if for any user in the demand matrix, the total cost of service demands generated by him equals to the total cost of service demands requested from him by other users in the demand set:*

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

LEMMA 2. *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 $C = \{u_0 \to u_1 \to \cdots u_m \to u_{m+1} = u_0\}$ be a loop in the demand graph, and $h^{(u_n,u_{n+1})} = h, 0 \le n \le m$. For $0 \le n \le m-1$, find a path $P_n$ in $G_S$ from $u_n$ to $u_{n+1}$. Obviously $P_{-m} \triangleq \bigcup_{0\le n\le m-1} P_n$ is a path (might with loops) from $u_0$ to $u_m$. Then $P_m \triangleq \{\langle j,i\rangle |$ 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 $\{(u_n, u_{n+1}), 0 \le n \le m\}$ as follows:

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

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

$$\sum_{n=0}^{m} x^{(u_n,u_{n+1})}_{\langle i,j\rangle} = \sum_{n=0}^{m} x^{(u_n,u_{n+1})}_{\langle j,i\rangle} \qquad \forall\langle i,j\rangle \in \bigcup_{0\le n\le m} P_n.$$

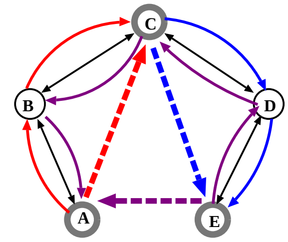



**Figure 4: Credit transfers for a balanced demand set**

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 4, 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 $A \to B \to C$ for service $(AC)$, $C \to D \to E$ for service $(CE)$, and path $E \to D \to C \to B \to A$ for service $(EA)$, all demands are executed, and the credit balances on all links remain zero.

LEMMA 3. *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 $u_i$, $0 \le i < n$, then we find a demand loop $u_i \to \cdots u_{n-1} \to u_n = u_i$. Let $h = \min_{i\le l\le n-1} h^{(u_l,u_{l+1})}$, 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 4. *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 3, we can decompose a balanced demand set into demand loops. According to Lemma 2, 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 credit limits do not constrain network trading because no peer has to run a "debt". In this case, NABT is perfectly efficient, that is, just as efficient as a global currency. 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.

## 4.4 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 section, we study the efficiency of NABT under unbalanced demands.

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 him does not equal to the total cost of service demands requested from him by other users 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*

$$\bar{H}(k,l) = H(k,l) - \min(H(k,l), H(l,k))$$

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 5. *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 4, 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). ■

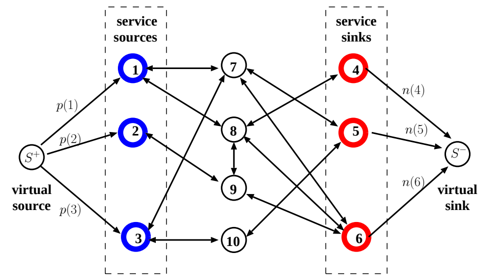



**Figure 5: Example of Extended Social Network**

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.*

For a social network with unbalanced demand matrix, we can build an extended social network as follows.

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:*

$$U' = U \cup \{s^+, s^-\}; \quad (4)$$
$$F' = F \bigcup_{u\in U^+} \{\langle s^+, u\rangle\} \bigcup_{v\in U^-} \{\langle v, s^-\rangle\}; \quad (5)$$
$$C_{\langle s^+,u\rangle} = p(u), \quad \forall u \in U^+; \quad (6)$$
$$C_{\langle v,s^-\rangle} = n(v), \quad \forall v \in U^-. \quad (7)$$

Figure 5 plots an extended social graph for an original social network graph with ten users, where three users are service sources, and 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 demand matrix.

THEOREM 6. *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 $\sum_{u\in U^+} p(u) = \sum_{v\in U^-} n(v)$.*

*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 $M$.

**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 $f(u,v), u \in U^+, v \in U^-$,

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

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

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

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

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 4, we can find credit transfer flows for $\tilde{H}$ without change the credit balance on links. Combine the credit flows for $\hat{H}$ and $\tilde{H}$, we have a credit flow for $\hat{H} + \tilde{H}$. It can be checked that the reduced demand matrix for $\hat{H} + \tilde{H}$ is the same as the reduced matrix for the original demand set $H$. Due to Lemma 5, $H$ is also executable. ■

In a trading system with a centralized bank and a global currency, service sinks can "buy" services as much as their balances with the bank allows. One can treat such a system as a trading network with a star topology rooted at the bank. The trading efficiency of 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.

# 5. NABT DESIGN CONSIDERATIONS

While we have shown NABT incentivizes and is highly efficient, practical design issues need to be addressed before it becomes a viable P2P incentive mechanism in online social networks. In this section, we provide a discussion on several NABT design considerations.

## 5.1 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.

### *5.1.1 Memoryless Property of Dynamic Routing*

We first show the memoryless property of credit routing.

THEOREM 7. *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

$\breve{A}(k-1)+A(k)$, which has the same reduced demand matrix as $H(k)$. According to Lemma 5, $H(k)$ 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. One may draw a conclusion that, at any given time $k$, all feasible routing solutions for the current demand set $H(k)$ are equally good for future routing. However, this argument is only true in terms of the routing feasibility. In reality, credit routing has to take many other considerations.

#### 5.1.2 Multiple Credit Routing Objectives

In the efficiency study, one can utilize any path to transfer credits and reach the maximum credit flow. In practice, long credit transfer paths are likely less preferable than shorter ones. Shorter credit transfer paths employ less number of trading intermediaries. It will increase the resilience of NABT against individual user failure and cheating. One has to simultaneously consider the "width" and "length" of credit transfer paths. Similar to multi-path traffic routing, it is also possible to employ multiple paths to transfer credits for a service demand requiring lots of credits. Since credit flow on each path affects the credit balance across each bilateral link, it affects subsequent incentives and the efficiency of asynchronous trading. For example, it might be better to "split up" one large credit flow between a particular demand and supply node into multiple flows across several disjoint paths connecting these two nodes. Such a protocol can make requests less "chunky" and hence improve efficiency by reducing the chance that bilateral links hit the credit constraints. Multipath credit routing can also increase the resilience against failures on individual paths. If a particular demand can be satisfied by more than one node, the routing protocol might "score" different possible paths by assessing how it affects intermediaries' credit constraints. We will investigate dynamic credit flow routing algorithms to tradeoff multiple credit routing objectives in our future work.

#### 5.1.3 Routing Calculation

Credit routing can be calculated by a centralized algorithm to maximally satisfy service demands. However, a centralized solution requires a greater degree of coordination and is hard to implement in online social networks. On the other hand, decentralized algorithms to find credit routing paths are closer to the spirit of asynchronous trading and can be easily adopted in online social networks. Similar to link-state routing, e.g. OSPF, in traffic networks, users in a social network can periodically exchange the status of their social links, including the connectivity, credit limit, credit balance. Using the collected network information, each user calculates credit routing for his demands locally. Users on the calculated routing path will then be notified to carry the credit transfer. Such a *proactive* approach might incur too much signalling overhead to exchange social link status. Alternatively, one can adopt a *reactive* approach: users calculate credit routing on-demand. After a user generates a service request, he will first search for potential service providers by sending out queries through the social network. [5] Then the user explore the credit transfer paths to those potential providers by sending out credit transfer requests through their neighbors. One simple way is to use *controlled flooding*. A credit transfer request with volume $h$ will first be sent to the user's direct neighbors with which it has an available credit budget no less than $h$ (that is, it can borrow additional $h$ credits without violating the credit limit constraint). Likewise, those neighbors will only forward the credit transfer request along the links with available credit space greater than or equal to $h$. When the credit transfer request reaches one of the supplier, it will send back a reply in the reverse direction to establish a credit transfer path. To control the overhead, one can adopt a *TTL* counter to limit the scope of flooding. In fact, as will be shown in our simulation of MySpace, even a two-hop flooding can already reach a large number of users in typical online social networks.

### 5.2 Dynamic Credit Limit Setting

In NABT, the credit limits $C_{ij}$ are individually set by the users and reflect the trust between friends in online social networks. As shown in Theorem 1, there is a tradeoff between efficiency and incentives. With dynamic service demands, users dynamically negotiate credit limits with their friends. Similar to practices in real social networks, a pair of friends tend to trust each other more as more trades between the two, direct or indirect, are fulfilled. And trust will be damaged by unfulfilled trades and service disputes. In NABT, users can adopt different policies for credit limit adjustment. In our simulations, 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 motivated to fulfill services requested from them and relay credits for their friends. Cheaters will be punished by losing their connectivity for future service trades. To avoid credit limit explosion, a maximum credit limit can also be set.

### 5.3 Additional Incentives for Intermediaries

In NABT, when relaying requests for service, intermediaries "break even" in trading bilateral credits. As explained in Section 3, the incentives for intermediaries come from that participating in credit transfers can maintain and strengthen their social links, through which they can obtain services in future. On the other hand, intermediaries do incur some small costs when relaying credits. Additional incentive mechanisms can be adopted to maximally motivate users to achieve

[5] It is also possible that service providers are located through a channel independent of the online social network.

a network-wide trading efficiency. One promising mechanism is to allow links to go "stale" if they haven't been used for relaying requests. This can be done by introducing a discount rate for credit limit on links. Essentially, more recent trades carry more weights in determining the current credit limit. To maintain enough credit limit for efficient trading, one is motivated to actively participate in NABT. Another mechanism is to compensate intermediaries by allowing them to charge a commission for relaying. This commission can be implicitly transferred by allowing the intermediary to swap credit with the requesting node for a slightly larger credit with the providing node.

# 6. SIMULATION STUDY OF NABT

To evaluate the performance of NABT, we developed a time-stepped simulator to compare NABT with synchronous trading and with trading with global currency. We drove the simulator with social network traces collected from MySpace [2]. Our simulation results highlight the incentives provided by NABT, the advantages of networked trading, and the impact of cheating behavior.

## 6.1 P2P Trading in File Sharing

We use P2P file sharing as an example to demonstrate the applicability and efficiency of NABT in online social networks. We simulate a P2P file sharing network with $10,000$ peers. Each peer dynamically generates requests to download files in a catalog of $10,000$ files. Each peer can share some files on its disk and upload to other peers. We assume peers are all self-optimizing and follow some strategy to trade files with other peers in the system. Three trading paradigms are investigated and compared:

**Trading with Global Currency (GCT).** We assume there is a global currency and a centralized bank. Peer $i$ has $B_i$ initial credits with the bank. For simplicity, we assume that each 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 to $j$ through the bank.

**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. To increase the trading opportunities, each peer locally buffers unfulfilled file download requests so that the request can be fulfilled in the future. Specifically, the trading policy is as follows: if a peer $i$ has a buffered request for a file that is available at another peer $j$, and at the same time peer $j$ has a buffered request for a file that is available at peer $i$, peer $i$ and $j$ exchange the files they want.

**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.

## 6.2 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.

### 6.2.1 File Profile

To simplify the simulation and focus on the P2P trading efficiency, we assume files are small and 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. When files are sorted in the descending order of popularity, the probability that the $j$th video is requested is $p_j = j^{-(1-\rho)}/I$, where $I$ is the normalization factor and $\rho$ is a control parameter. We chose $\rho = 0.27$, which is a commonly used factor for video on-demand services [6].

### 6.2.2 Peer Profile

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 [17]. 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 reenters 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 $\gamma$. The default requesting rate is set to 1/10. Thus, on average, every 10 minutes, a peer requests a new file. The file requested again follow the Zipf distribution. After a peer generates a new file request, it inserts the request into its *request queue*. Meanwhile, a peer maintains *serve queue* to buffer the received requests from other peers. Requests in the serve queue are served in a first-in-first-out (FIFO) fashion. Within each time-slot, each peer attempts to find supplying peers that can fulfill the requests in its request queue. Different systems have different trading policies to assign these requests. For a particular request, when there is more than one peer that satisfies a given trading policy, the peer will select a supplying peer that can serve this file with the shortest waiting time. It is also possible that a peer cannot find one supplying peer for a particular request. In this case, the peer will eave this request in its request queue, and try again in the next time-slot. If a file request cannot be sched-

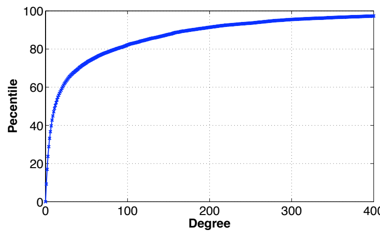



**Figure 6: CDF of the number of friends in the social network topology from MySpace.**

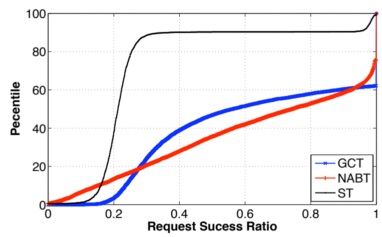



**Figure 7: Trading efficiency for different systems: CDF of request success ratio**

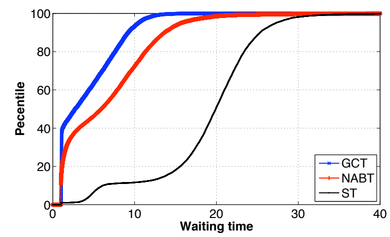



**Figure 8: Trading efficiency for different systems: CDF of waiting time**

uled within 60 minutes after it has been generated, the peer will simply *drop* this request and remove it from its request queue. We assume the request queue of each peer has a finite length (in our simulation, 100 requests). If a peer's request queue is full, all newly generated requests are *blocked*.

### *6.2.3 Trading Configuration*

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 [2]. The crawler randomly selects a starting user, crawls the user's friends' pages, their friends' pages, and so on. Figure 6 shows the CDF of the number of friends in this social network topology. On average, a user has 127 friends. 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 lines according to the AIMD algorithm described in Section 5.2, 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 intitial pairwise credits in NABT with all of its active friends. Thus, $B_i$ is set to be $C * T_{on}/(T_{on} + T_{off}) * D_i$, where $D_i$ is the number of friends of peer $i$ in NABT.

## 6.3 Simulation Results

### *6.3.1 Trading Efficiency Comparison*

We compare the trading efficiency of the three paradigms mainly using two performance metrics:

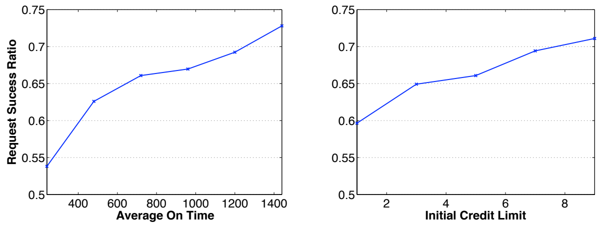
- **Request success ratio ($\theta$):** The ratio of fulfilled requests to the total number of requests for a peer. The canceled and blocked requests are not counted.

- **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 the delay in the request queue on the requesting peer and the delay in the serve queue in the supplying peer.

Figure 7 and 8 show the CDF of $\theta$ and $T$ across all peers for the NABT, ST, and GCT schemes. From Figure 7, we observe that the ST system generally has a low request success ratio. About $90\%$ of peers have a $\theta$ lower than 0.25. 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 $\theta$. The average request successs ratio for the content-rich peers is as high as 0.97. From Figure 8, we can see that the ST system has a long waiting time $T$ given that a request can be successfully scheduled. More than $50\%$ of peers have an average waiting time longer than 20 minutes. 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 waiting time in the request queue at a requesting peer, 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.

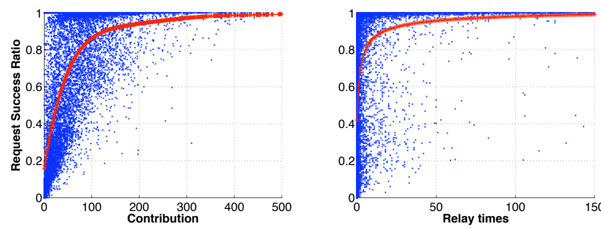
The GCT system has a much higher request success ratio, as indicated in Figure 7. More than $40\%$ of peers can successfully schedule all of their file requests. These peers have enough credits to pay for all of the files they want. They either have a large amount of initial credits, or earn sufficient 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 8 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. The delay in the request queue is small.

Figure 7 and 8 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




(a) Average peer on time (b) Initial credit line

**Figure 9: Impact of system configurations on the request success ratio in NABT.**




(a) upload contribution (b) relay contribution

**Figure 10: Relation between request success ratio and user contribution**

pass the credits to the supplying peer. For peers not well connected in the social network, their file requests may need to wait in the request queue for a while to obtain an available path to pass the credits. As a result, the waiting time of NABT is slightly longer than GCT.

#### 6.3.2 *Sensitivity of System Configurations*

We investigate the sensitivity of system configurations, such as the number of on-line users and the initial credit limits, on the trading efficiency of NABT. Figure 9(a) shows the average request success ratio over all peers as a function of the average on time $T_{on}$. We fix the average off time $T_{off}$ to 720 minutes. Thus, a larger $T_{on}$ leads to more on-line peers. As expected, $\theta$ increases as peers stay longer in the system. With more on-line peers, it is easier for a peer to find credit transfer paths. The initial credit limit is another design parameter that affects the trading efficiency for the NABT system. As indicated in Figure 9(b), with a higher initial credit limit, peers can use more credits to trade with each other, hence increasing the trading efficiency.

#### 6.3.3 *Service Differentiation*

A good incentive design leads to service differentiation among peers with different service contributions. Figure 7 and 8 show that different peers have different success ratios and waiting times. In this section, we will see whether these differentiated services relate to the peers' contributions to the system. Figure 10(a) relates the success ratio of each peer to its upload contribution (in terms of number of files).

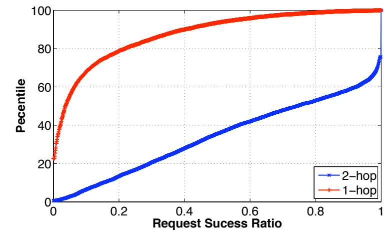



**Figure 11: CDF of request success ratio for bilateral trading with and without intermediaries.**

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

#### 6.3.4 *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 11 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. $70\%$ of the peers have request success ratio lower than $0.1$. 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.

#### 6.3.5 *Impact of Cheating*

We now consider free-riding and cheating behavior. In

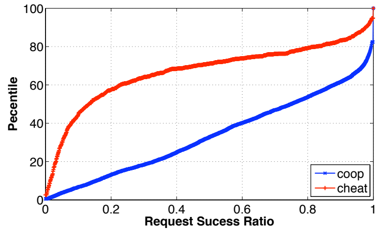



**Figure 12: CDF of request success ratio for NABT with the presence of free-riders.**

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 lines 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 a probability of 0.5. We investigate whether the system can provide a limited service quality to the free-riders. Figure 12 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, the free-riders receive much worse service compared with the cooperative peers: more than 60% of free-riders have a $\theta$ lower than 0.2. The cooperative peers have a similar $\theta$ as when there are no free-riders (Figure 7). 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.

## 7. RELATED

As mentioned in the Introduction, there are several studies on BitTorrent's tit-for-tat incentive mechanism [8, 23, 10, 24, 25, 31, 33]. As described in Section 1, BitTorren'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 [35, 34, 4, 14]. 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 [18, 16]; 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 [26, 11, 12]. Feldman et al use simulation to show that reciprocation incentive mechanisms can drive the system of strategic users to nearly optimal levels of cooperation [12]. More recently, Zhao et al developed a mathematically-tractable dynamic game-theoretic framework to analyze a broad class of P2P incentive schemes [36]. To our knowldege, this paper, proposing and exploring NABT, is the first to examine P2P incentives in the context of online social networks.

## 8. CONCLUSIONS AND FUTURE WORK

We presented a new P2P trading paradigm for online social networks: Networked Asynchronous Bilateral Trading (NABT). NABT solves the two fundamental problems of traditional synchronous bilateral P2P trading: asynchronicity over time and asychronicity over nodes. In NABT, peers can trade services with each other both asynchronously and through intermediaries. Supporting a form of "network tit-for-tat" we mathematically showed that NABT is perfectly efficient with balanced demands. The efficiency of NABT with unbalanced demands is determined by the min-cut between service sources and service sinks. We discussed practical design considerations in dynamic trading environment. To demonstrate the efficiency of NABT, we developed two-hop NABT algorithms for P2P file sharing systems. 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.

NABT opens up an exciting space for P2P incentive design. An important open problem is optimal credit routing algorithms to tradeoff multiple routing objectives. It is desirable to design dynamic credit limit adjustment algorithms to adapt to user demands and user trading behaviors. It is also desirable to develop additional incentive strategies that maximally motivate intermediaries to participate in networked trading. Another important direction for future work is to implement the NABT algorithms as incentive engines for P2P applications in online social networks. The implementation of NABT is straightforward and only requires minimum coordination among friends in social networks.

## 9. REFERENCES

[1] ADAR, E., AND HUBERMAN, B. A. Free Riding on Gnutella. *First Monday 5* (2000).

[2] AHN, Y.-Y., HAN, S., KWAK, H., MOON, S., AND JEONG, H. Analysis of topological characteristics of huge online social networking services. In *WWW* (2007).

[3] ANDERSON, D., AND KUBIATOWICZ, J. The Worldwide Computer. *Scientific American 286(3)* (2002).

[4] APERJIS, C., FREEDMAN, M. J., AND JOHARI, R. Peer-assisted content distribution with prices. In *Proceedings of CoNext* (2008).

[5] BIAN, Y. *Getting a Job Through a Web of Guanxi in China*. Networks in the Global Village: Life in Contemporary Communities, Westview Press, 1999.

[6] CHU, J., LABONTE, K., AND LEVINE, B. N. Availability and locality measurements of peer-to-peer file systems. In *ITCom* (2002).

[7] CLARK, T. Digicash files chapter 11. http://news.cnet.com/2100-1001-217527.html.

[8] COHEN, B. Incentives build robustness in BitTorrent. In *Workshop on Economics of Peer-to-Peer Systems* (June 2003).

[9] COLEMAN, J. S. *Foundations of Social Theory*. Harvard University Press, 1990.

[10] FAN, B., CHIU, D.-M., AND LUI, J. C. S. The delicate tradeoffs in BitTorrent-like file sharing protocol design. In *IEEE ICNP* (November 2006).

[11] FELDMAN, M., LAI, K., STOICA, I., AND CHUANG, J. Robust incentive techniques for peer-to-peer networks. In *EC* (2004).

[12] FELDMAN, M., PAPADIMITRIOU, C., CHUANG, J., AND STOICA, I. Free-riding and whitewashing in peer-to-peer systems. In *Workshop on Practice and theory of incentives in networked systems* (2004).

[13] GARICANO, L., AND SANTOS, T. Referrals. *American Economic Review* (2004).

[14] GOLLE, P., LEYTON-BROWN, K., AND MIRONOV, I. Incentives for sharing in peer-to-peer networks. In *ACM Conference on Electronic Commerce 2001* (2001).

[15] GRANOVETTER, M. *Getting a Job*. Harvard University Press, 1974.

[16] GUPTA, M., JUDGE, P., AND AMMAR, M. A reputation system for peer-to-peer networks. In *Proceedings of NOSSDAV* (2003).

[17] HUANG, C., LI, J., AND ROSS, K. W. Can Internet VoD be profitable? In *ACM SIGCOMM* (Kyoto, August 2007).

[18] KAMVAR, S. D., SCHLOSSER, M. T., AND GARCIA-MOLINA, H. The eigentrust algorithm for reputation management in P2P networks. In *Proceedings of WWW* (2003).

[19] KANDORI, M. Social norms and community enforcement. *Review of Economic Studies* (1992).

[20] KARLAN, D., MOBIUS, M., ROSENBLAT, T., AND SZEIDL, A. Trust and social collateral. *To appear in Quarterly Journal of Economics* (2008).

[21] KARLAN, D., MOBIUS, M., SZEIDL, A., AND ROSENBLAT, T. Trust and social collateral. *forthcoming in Quarterly Journal of Economics* (2008). `http://mobius.fas.harvard.edu/papers/socialcollateral.pdf`.

[22] KRANTON, R. E. Reciprocal Exchange: A Self-Sustaining System. *American Economic Review* (1996).

[23] LEGOUT, A., LIOGKAS, N., KOHLER, E., AND ZHANG, L. Clustering and sharing incentives in BitTorrent systems. In *ACM SIGMETRICS* (June 2007).

[24] LIOGKAS, N., NELSON, R., KOHLER, E., AND ZHANG, L. Exploiting BitTorrent For Fun (But Not Profit). In *Proc. of IPTPS* (2006).

[25] LOCHER, T., MOOR, P., SCHMID, S., AND WATTENHOFER, R. Free riding in BitTorrent is cheap. In *ACM HotNets 2006* (Irvine, November 2006).

[26] MA, T. B., LEE, C. M., LUI, J. C. S., AND YAU, K. Y. Incentive and service differentiation in P2P networks: A game theoretic approach. *IEEE/ACM Trans. on Networking 14*, 5 (October 2006).

[27] MAS-COLELL, A., WHINSTON, M. D., AND GREEN, J. R. *Microeconomic Theory*. Oxford University Press, 1995.

[28] MCCOY, J. Mojo nation responds. http://pragmatic.oreilly.com/pub/a/p2p/2001/01/11/mojo.html.

[29] MOBIUS, M. Trading favors. *Harvard University Working Paper* (2001). `http://mobius.fas.harvard.edu/papers/favors.pdf`.

[30] MOBIUS, M., AND SZEIDL, A. Trading favors ii. `http://mobius.fas.harvard.edu/papers/favors2.pdf`.

[31] PIATEK, M., ISDAL, T., ANDERSON, T., KRISHNAMURTHY, A., AND VENKATARAMANI, A. Do incentives build robustness in bittorrent. In *Proc. of NSDI* (2007).

[32] SAROIU, S., GUMMADI, P. K., AND GRIBBLE, S. D. A measurement study of Peer-to-Peer file sharing systems. In *Proc. of Multimedia Computing and Networking (MMCN)* (2002).

[33] SIRIVIANOS, M., PARK, J. H., CHEN, R., AND YANG, X. Free-riding in BitTorrent networks with the large view exploit. In *IEEE IPTPS* (February 2007).

[34] VISHNUMURTHY, V., CHANDRAKUMAR, S., AND SIRER, E. G. Karma: A secure economic framework for peer-to-peer resource sharing. In *Proceedings of P2P-Econ* (2003).

[35] YANG, B., AND GARCIA-MOLINA, H. PPay: Micropayments for peer-to-peer systems. In *Proceedings of CCS* (2003).

[36] ZHAO, B. Q., LUI, J. C. S., AND CHIU, D.-M. Mathematical modeling of incentive policies in P2P systems. In *NetEcon* (2008).