Reciprocity in Peer-to-Peer Systems

Daniel Sadoc Menasché •, Laurent Massoulié ⋆, Don Towsley •
⋆Thomson Research, •University of Massachusetts at Amherst

Abstract—**This work investigates reciprocity in peer-to-peer systems. The scenario is one where users arrive to the network with a set of contents and content demands. Peers exchange contents to satisfy their demands, following either a direct reciprocity principle (*I help you and you help me*) or indirect reciprocity (*I help you and someone helps me*). First, we prove that any indirect reciprocity scheduling of exchanges can be replaced by a direct reciprocity one, provided that users *(1)* are willing to download undemanded content for bartering purposes and *(2)* use up to twice the bandwidth they would use under indirect reciprocity. Motivated by the fact that the loss of efficiency due to direct reciprocity is at most two, we use simulations to study various distributed direct reciprocity schemes, some of them involving a broker to facilitate exchanges.**

I. INTRODUCTION

Reciprocity is one of the fundamental pillars supporting peer-to-peer systems. In essence, the principle of reciprocity states that participants must contribute to the system with resources such as bandwidth or memory in order to accomplish their tasks. The primary goals of this paper are to present new foundational understanding about reciprocity in peer-to-peer systems, and to show how practical design decisions used to incentivize reciprocity impact performance.

Our work is motivated by systems in which users exchanging contents incur almost zero costs to replicate those contents. Examples are peer-to-peer swarming systems such as BitTorrent and content trading systems such as TitleTrader [1]. In the former, users demand files and exchange chunks, while in the latter users demand and exchange non splittable commodities such as DVDs. Henceforth, we focus on peer-to-peer swarming systems such as BitTorrent.

In BitTorrent reciprocity happens naturally between users interested in the same content at the same time. Those users join a swarm and exchange chunks of files among themselves. But swarming systems also support reciprocity between users interested in different contents. That is because a swarm may be associated with a bundle of files [2]. In this case, users joining a swarm can download only a subset of the available files and may already own some files when arriving to the system. In addition, users may download more content than they originally sought, for bartering purposes.

Reciprocity mechanisms broadly classify into two types. In the case of direct reciprocity, users follow the principle of *I scratch your back and you scratch mine*. In the case of indirect reciprocity, the return may not come necessarily from the recipient, i.e., peer A provides content to a peer B at a rate equal to that at which content is being provided to it by peer C, regardless of whether B and C are the same individual. Users follow the principle of *give and you shall be given* [3]. Both direct and indirect reciprocity are used in peer-to-peer systems.

BitTorrent implements a direct reciprocity tit-for-tat incentive mechanism. The mechanism maintains a long term exchange between two peers only if they both benefit from it. Even though BitTorrent does not have the strict requirement that the two peers exactly match each other's flows rates, it has been suggested that imposing this requirement leads to higher robustness [4] and BitTyrant already employs it [5].

Indirect reciprocity is implemented in peer-to-peer systems such as eMule [6] and PACE [7] that rely on credits to accomplish fair exchanges. Other systems such as PeerTrust [8] take advantage of indirect reciprocity through reputation mechanisms. In some cases indirect reciprocity may be the only feasible solution [9], [10]. Nevertheless, in the realm of peer-to-peer content distribution, direct reciprocity has its own advantages.

Direct reciprocity systems are simpler to implement than their credit based counterparts. In credit based systems users need to either rely on a bank (single point of failure) or store information in local files about credits received or given (prone to hacking). Inspired by the simplicity of direct reciprocity, we ask the following two questions,

- what is the loss of efficiency for enforcing direct rather than indirect reciprocity?
- how can trackers help users achieve efficient direct reciprocity schedules?

To answer the first question, we formulate a model of users with content and content demands. Using this model, we show that any indirect reciprocity schedule can be replaced by a direct reciprocity schedule, provided that *(1)* users are willing to obtain undemanded content for the purpose of barter and *(2)* they are willing to use up to twice the bandwidth resources that they would have used under indirect reciprocity.

To address the second question, we propose a *broker-based* architecture (consider a broker to be a sophisticated BitTorrent tracker) and study, through simulation, several schemes where the broker provides greater amounts of information to users for the purpose of understanding the value of this information.

Our key contributions are the following,

- we show that the loss of efficiency due to direct reciprocity, measured in terms of number of transmissions made by each user, is at most two;
- we propose mechanisms based on brokers to perform match-making between users and to issue recommendations on content value for bartering. We validate experimentally how they enable efficient direct reciprocity exchanges.

The remainder of this paper is organized as follows. In Section II we show our main result concerning the loss of efficiency due to direct reciprocity. Sections III and IV provide the system design and experimental results. Section V discusses the related literature and Section VI concludes.

II. LOSS OF EFFICIENCY DUE TO DIRECT RECIPROCITY

In this section we prove that the loss of efficiency due to direct reciprocity is at most two. To appreciate the nature of our result in a simple setting, consider the cycle shown in Figure 2(a), in which node v owns content c_v and requests content c_{v-1}, $0 \leq v \leq 2$, with subtraction done modulo 3. Under an indirect reciprocity constraint, every node transmits one content so as to have all demands satisfied. Beginning with the same workload, we now seek a direct reciprocity schedule. If *(a)* nodes are willing to receive one additional content for bartering purposes and *(b)* at least one node can transmit two contents simultaneously, one such schedule is shown in Figure 2(b). In this schedule the loss of efficiency, measured as the ratio of number of transmissions by a node in the direct and indirect reciprocity schedules, is two. In the schedule shown in Figure 2(c) the loss of efficiency is 4/3.

Clearly, the loss of efficiency due to direct reciprocity is no more than two for any cycle. The direct reciprocity schedule consists of transmitting all contents in clockwise direction, except for one content which flows counterclockwise. In what follows, we show that the loss of efficiency of two extends from cycles to any configuration.

A. Model

We define a network to be $N = (V, C, S, D, E)$ where V is a finite vertex set, C is a set of contents, S identifies the set of sources for each content, $S : C \to 2^V$, D identifies the set of contents required by each node,[1] $D : C \to 2^V$, and $E \subset V^2 \times C \times \mathbb{R}^+$ (Table I contains the notation used in the paper).

If $e = (v_1, v_2, c, r) \in E$, then there exists a flow of content c from $v_1 = t(e)$ to $v_2 = h(e)$ at rate r. We use the notation $(t(e), h(e), c(e), r(e)) \in E$ to represent such an element. Note that unlike a directed graph, there may be several edges carrying different flows at different rates from one node to another. We assume that when a node starts transferring content to another node, that content shows up immediately at the second node.

variable	description
V	set of users
C	set of contents
$S : C \to 2^V$	set of sources for each content
$D : C \to 2^V$	set of users that demand each content
$E \subset V^2 \times C \times \mathbb{R}^+$	set of labeled edges; each edge e transmits
$\{e = (v_1, v_2, c, r)\}$	content c from v_1 to v_2 at rate r
$N(V, C, S, D, E)$	dissemination network
$p_N(j)$	incoming edges to user j in network N
$s_N(j)$	outgoing edges from user j in network N
C_v	contents initially owned by v (v's endowment)
D_v	set of contents initially demanded by user v
t_v^i	transmissions by user v under indirect reciprocity
t_v^d	transmissions by user v under direct reciprocity
L	(user) loss of efficiency, $\max_{v \in V} t_v^d / t_v^i$
$\overline{L}$	system loss of efficiency, $(\sum_{v \in V} t_v^d)/(\sum_{v \in V} t_v^i)$

TABLE I
TABLE OF NOTATION.

Let C_v denote the set of contents that node v initially has, i.e., $C_v = \{c : v \in S_c\}$. C_v is also referred to as node v endowment. C_v^* denotes the set of contents available to that node $v \in V$, either because it is a source of that content or because it receives it; $C_v^* = \{c : v \in S_c\} \cup \{c : c(e) = c \quad \forall e \in E \text{ s.t. } h(e) = v\}$.

Let $p_N(j)$ denote the set of incoming edge flows to j in network N, $p_N(j) = \{e \in E : h(e) = j\}$. $p_N(S)$ is the set of incoming edge flows to set $S \subset V$. Similarly, $s_N(i)$ (resp., $s_N(S)$) is the set of outgoing edge flows from $i \in V$ (resp. $S \subset V$). Let $E(S_1, S_2)$ denote the set of edges directed from nodes in S_1 to nodes in S_2.

We say that a network is a *dissemination network* if

1) for every node-demand pair (v, c), $v \in V$, $c \in D_v$, there exists a path from a source of content c to v consisting of edge flows carrying solely content c;
2) nodes don't receive redundant content over different edges.

A dissemination network is an *indirect reciprocity network* if

$$\sum_{e \in p(i)} r(e) = \sum_{e \in s(i)} r(e), \quad \forall i \in V,$$

and a *direct reciprocity network* if for all $i, j \in V$,

$$\sum_{e \in E} r(e) \mathbf{1}\{t(e)=i, h(e)=j\} = \sum_{e \in E} r(e) \mathbf{1}\{t(e)=j, h(e)=i\},$$

where $\mathbf{1}\{\mathcal{P}\}$ is an indicator variable equal to 1 if $\mathcal{P}$ is true.

B. Main Result

Let the loss of efficiency due to direct rather than indirect reciprocity, L, be defined as the maximum of the ratio of

---

[1] We use the terms *node* and *user* interchangeably.

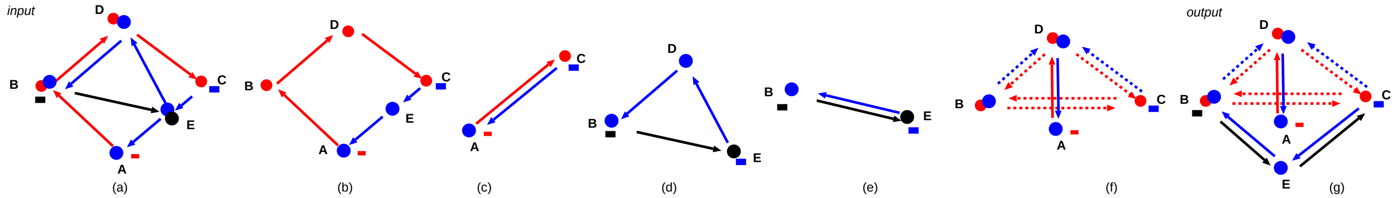

Fig. 1. Illustrating the proof: (a) input; (b) first removed cycle and (c) its corresponding logical direct reciprocity network; (d) second removed cycle and (e) its corresponding logical direct reciprocity network; (f) direct reciprocity network [from (c) after introducing B and D]; (g) direct reciprocity output.

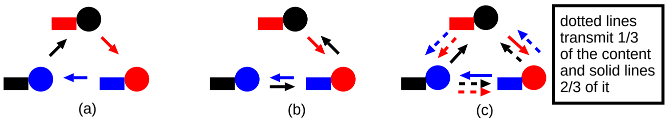

Fig. 2. (a) Indirect reciprocity cycle; (b) direct reciprocity with loss of efficiency of two; (c) direct reciprocity with loss of efficiency of 4/3.

number of transmissions made by a node in a direct reciprocity schedule, t_v^d, over the number of transmissions under an indirect reciprocity schedule, t_v^i,

$$L = \max_{v \in V} t_v^d / t_v^i \tag{1}$$

Next, we prove that given an indirect reciprocity network there is always a direct reciprocity one such that $L \leq 2$,

Theorem 2.1: The loss of efficiency due to direct reciprocity is at most 2.

C. Proof Overview

The proof of the main result is based on repeatedly transforming cycles in an indirect reciprocity dissemination network N that do not satisfy direct reciprocity into sets of direct reciprocity cycles without doubling the flow rate in/out of a node. We execute the following algorithm.

Algorithm 1 INDIRECTTODIRECT

```
Input: indirect reciprocity network $N = (V, C, S, D, E)$
Output: direct reciprocity network $N'$
1: $i \leftarrow 0$
2: while $E \neq \emptyset$ do
3:     $i \leftarrow i + 1$
4:     remove a cycle $A_i$ from $N$ (see §II-E.1)
5:     reduce $A_i$ to a logical direct reciprocity dissemination network $N_i$ (see §II-E.2)
6:     consider all branching points of contents involved in $A_i$ to now be sources of corresponding content
7: end while
8: convert the set of logical direct reciprocity dissemination networks $\{N_j, j = 1, \ldots, i\}$, into a direct reciprocity dissemination network $N'$ (see §II-E.3)
```

Our algorithm is constructed so as to satisfy two key properites,

- **efficiency**: the number of transmissions by each node in the direct reciprocity network is at most twice the number in the indirect reciprocity network;
- **correctness**: the direct reciprocity network is valid, i.e., for every node-demand pair (v, c) there exists a path from a source of c to v consisting of edge flows carrying c.

When studying the later we will focus much of our attention in showing that all content that reaches a node ultimately comes from a source, i.e., the graph characterizing the distribution of each content is loop free.

D. Illustrative Example

Let us now present a simple example to illustrate how Algorithm 1 (detailed in the next section) works and why it satisfies the two desired properties. The example has only five nodes and three contents (Figure 1) but brings out the key insights in the conversion from indirect to direct reciprocity.

The first step consists of removing a cycle from the indirect reciprocity network (line 4 in Algorithm 1). This is illustrated in Figure 1(b). Note that the sources of all contents that flow in the cycle are present. In §II-E.1 we show that it's always possible to find such cycle.

The second step consists of short-circuiting consecutive edges of the cycle that transmit the same content. This new cycle is referred to as a logical indirect reciprocity cycle (see Figure 1(c)). In our simple example, the logical indirect reciprocity cycle is already a logical direct reciprocity cycle. In §II-E.2 we show that it's always possible to convert the former into the later (line 5 in Algorithm 1).

In the remaining indirect reciprocity network, we mark node E, a branching point for content blue, as a new source for that content (line 6 in Algorithm 1). We then repeat the steps above to obtain the networks in Figures 1(d) and 1(e). Note that when generating the logical direct reciprocity networks each node is assumed to have only the contents that it already acquired (endowment plus contents received from other nodes). Therefore, once the demand for a content is satisfied, the demanded content is genuinely issued at least once. The later is a necessary condition for the correctness property to hold.

Finally, in §II-E.3 we show that it's always possible to group the obtained set of logical direct reciprocity networks into a direct reciprocity network (line 8 in Algorithm 1) while satisfying the efficiency property. This is accomplished by intercepting, in the logical direct reciprocity cycles, the nodes that still need to receive contents (see Figure 3), and then combining the resulting networks together. In our example, we intercept nodes B and D in the network in Figure 1(c) to obtain the one shown in Figure 1(f). Combining the networks shown in Figures 1(e) and 1(f) yields the desired output (see Figure 1(g)).

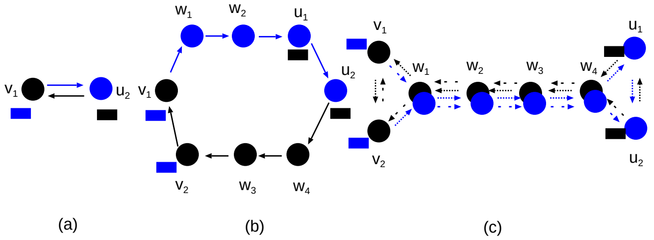

Fig. 3. Coping with logical flows, (a) one logical direct reciprocity cycle with two logical flows, (b) corresponding indirect reciprocity cycle, (c) a dumbbell graph, where direct reciprocity holds.

E. Proof Details

Next, we provide the remaining details of the proof.

1) ***Removing Cycles from the Indirect Reciprocity Network:*** To remove cycles from the indirect reciprocity network we proceed as follows. First, pick the source of any content. It is easy to show that sources for a content must also play the role of leaves for other contents. Proceeding backwards, identify content that the source is a leaf of, and corresponding source. Proceed until a cycle is found. In this cycle, the sources of all contents that flow in the cycle are present.

To reduce a cycle into a direct reciprocity network, we first replace in the cycle a sequence of edge flows with the same content (Figure 1(b)) by a single *logical edge flow* for that content (Figure 1(c)). The resulting cycle is a *logical simple cycle*. We then reduce the logical simple cycle into a logical direct reciprocity dissemination network.

2) ***Reducing Simple Cycles to Logical Direct Reciprocity Networks:*** The key insight of our reduction is to relate simple cycles to perfect matchings in bipartite graphs. Let $G = (V, C, S_1, E_1)$ be a simple cycle in a dissemination network N. Now suppose that U is a copy of V and let $B = (V \cup U, E \subseteq V \times U)$ denote a bipartite graph representing possible content exchanges between nodes in the cycle. Edge $(v_i, v_j) \in E$ if v_i is a potential source for v_j in the dissemination network G. With a slight abuse of notation, $(v_i, v_j) \in E$ if $c_{v_j} \in s_G(v_i)$.

A matching on B consists of a subset $M \subseteq E$ such that each node in $V_1 \cup V_2$ is incident to at most one edge in M. A *perfect matching* is one where each node in V_1 is incident to exactly one edge in M. Clearly, given any cycle in N we can construct a corresponding perfect matching in B, and vice versa. An edge $(v_i, v_j) \in E$ corresponds to node v_j being the sucessor of v_i in cycle G.

Our result on cycle reductions relies on the following lemma about perfect matchings, the proof of which relies on Hall's theorem [11, Thm. III.7] and can be found in the Appendix. Letting $s_B(u)$ be the number of incident edges to vertex u in a bipartite graph B, we have

Lemma 2.1: Consider a bipartite graph $B = (V, U, E)$ where $V = \{v_1, \ldots, v_n\}$ and $U = \{u_1, \ldots, u_n\}$ that satisfies the following, *(i)* $(v_i, u_i) \in E$, $i = 1, \ldots, n$, and *(ii)* $s_B(u) \geq 2$, $u \in U$. There exist at least two perfect matchings M_1, M_2 such that $M_1 \neq M_2$ whenever $n \geq 3$.

We now focus on reducing a simple cycle to a set of direct reciprocity cycles without more than doubling the bandwidth requirement.

Lemma 2.2: Any simple cycle can be converted into a direct reciprocity network incurring a loss of efficiency of at most two.

Proof. We consider two cases.

Case i) There exists at least one content that can only be provided by one node. This content can be used for bartering purposes and it is easy to see that in this case the lemma holds (see conversion from Figure 2(a) to Figure 2(b)).

Case ii) Every content can be provided by at least two nodes. We describe a transformation that shifts the amount of content delivered over indirect reciprocity cycles to direct reciprocity cycles. In particular, this transformation moves a minimum of $3r/2^{n-2}$ of additional flow to direct reciprocity cycles.

Construct a bipartite graph $B = (V, U, E)$ where V and U are copies of V_a, and $E = \{(v_i, v_{i+1})\} \cup \{(v_i, v_j) : c_{v_j} \in s(v_i)\}$. By Lemma 2.1, there exist two perfect matchings M_1 and M_2 such that $M_1 \neq M_2$, one of which, M_1, corresponds to the content flows in A. Using these matchings, we transform A into three or more cycles where each one serves half of each content at half of the rate. Moreover, at least one cycle contains less than n nodes. Algorithm 2 is used to produce these cycles. A sample input and a sample output are illustrated in Figure 4.

Algorithm 2 halts because every node appears once as the tail of an edge and once as the head of an edge in each of M_1 and M_2. This ensures that a node appears in exactly two cycles. In addition, the first cycle, A_1 contains fewer than n nodes. We construct the network dissemination cycles by

```
Algorithm 2 CYCLEREDUCE
Input: logical indirect reciprocity cycle $A = (V_a, C_a, S_a, E_a)$
    and matchings $M_1$ and $M_2$
Output: logical indirect reciprocity network
initialize: split contents $c_i$ into $c_i^1$ and $c_i^2$
choose an arbitrary edge $e \in M_2$ such that $e \notin M_1$
$M_2 = M_2 \setminus \{e\}$; $V_1 = \{t(e)\}$; $E_1 = \{e\}$
$i = 1$
while $M_1$ or $M_2$ is not empty do
    {construct cycle $A_i = (V_i, E_i)$ as follows}
    while $h(e) \notin V_i$ do
        select $e'$ from $M_1$ or $M_2$ such that $t(e') = h(e)$
        insert $h(e)$ into $V_i$ and $e'$ into $E_i$
        remove $e'$ from the appropriate matching
        $e = e'$
    end while
    identify cycle $A_i$
    re-add edges in $E_i$ not in $A_i$ into original matchings
    if $M_1$ or $M_2$ is not empty then
        $i = i + 1$
        choose arbitrary edge $e$ from $M_1$ or $M_2$
        remove $e$ from the appropriate matching
        $V_i = \{t(e)\}$, $E_i = \{e\}$
    end if
end while
```

associating $c_{v_i}^1$ to the edge on which v_i is the head in one of the cycles containing v_i and $c_{v_i}^2$ to the other such edge and a rate of $r/2$ to each edge. If either A_1 is a direct reciprocity cycle or satisfies Case *i)*, we are done. Otherwise repeat this process on A_1 and the smallest cycle that is produced at each step. This produces a cycle of size three within $n-2$ steps. It is trivial to reduce this three node cycle to direct reciprocity cycles. This increases the flows carried by direct reciprocity cycles by at least $3r/2^{n-2}$. Note that at the end of this repeated application of the cycle-reduce algorithm, we are left with a set of simple cycles carrying some amount of flow. We can now apply Case i) and Case ii) to these cycles with the consequence that flow will continue to be shifted over to direct reciprocity cycles. This can be repeated until all flow is being carried over direct reciprocity cycles.

Two questions remain: *(1)* Do these transformations result in all content flows being shifted from indirect reciprocity cycles to direct reciprocity cycles? *(2)* Is the bandwidth required by the direct reciprocity flows no more than double that required by the original cycle? The answer to *(1)* is *yes* as any indirect reciprocity cycle can be reduced by repeated application of the above algorithm resulting in a positive decrease in indirect reciprocity flow. The answer to *(2)* is *yes* based on the following argument. The algorithm above replaces cycles by a set of smaller cycles; however the per node bandwidth requirements remain unchanged. Bandwidth requirements are only doubled (and in only some cases) at the last step when a direct reciprocity cycle is created. ■

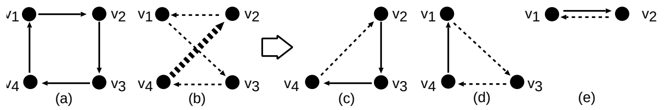

Fig. 4. Illustration of the cycle-reduce algorithm: the input to the algorithm is a cycle C and two matchings, (a) M_1 and (b) M_2; the first edge selected by the algorithm, $e = (v_4, v_2) \in M_2 \setminus M_1$, is marked in bold; (c) the first cycle constructed by the algorithm, A_1, has fewer nodes than C; e is the only edge from M_2 in this cycle; (d) the second cycle, A_2, and (e) the third cycle, A_3. Every node is in two cycles constructed by the algorithm. Hence, since each cycle transfers half of the contents in C, nodes receive the same amount of content in C as in the superposition of A_1, A_2 and A_3.

Note that our result relies on fractional allocations. If nodes can't split their traffic among multiple paths, the loss of efficiency may be more than two (see Appendix).

*3) **From Logical Cycles to a Direct Reciprocity Network**:* The final step is to satisfy the demands of the nodes that are not encompassed by logical direct reciprocity cycles, and to combine these cycles into a single direct reciprocity network. To this aim, we replace each *(a)* logical direct reciprocity cycle by its corresponding *(b)* indirect reciprocity cycle and reduce the resulting cycle into a *(c)* set of direct reciprocity cycles, see Figures 3(a), 3(b) and 3(c), respectively.

Given a logical direct reciprocity cycle involving contents a and b, we divide the nodes that do not appear in this cycle (but that demand a or b) into four sets according to whether they demand content a or b and whether they have content a or b available. Let V_c be the set of deprived nodes that neither have a nor b but demand $c \in \{a, b\}$ and let $V_{a,b}$ (resp. $V_{b,a}$) denote the set of nodes that demand a (resp. b) and have b (resp. a). Once the nodes are classified in these four sets, it is easy to see how to include them into the logical direct reciprocity network preserving direct reciprocity as shown in Figure 3(c) (see also [12]). After incorporating all nodes into corresponding logical cycles and all nodes have their demands satisfied, the resulting network is the desired direct reciprocity network.

In Figure 3(c), note that nodes $w_1, \ldots, w_m$ utilize twice the bandwidth that they previously needed for this content. However, this is just for this flow. Thus it remains the case that the node bandwidth requirement never more than doubles for each content flow a node was originally involved with. Hence, the total node bandwidth never more than doubles. ■

F. Tightness of the Bound

To show that the bound of two on the loss of efficiency is tight, consider a cycle of size $|V|$. Each node v owns content

c_v and demands c_{v-1}, with subtraction done modulo $|V|$. Let $T_{|V|}$ be the minimum aggregate number of transmissions in a direct reciprocity schedule for a cycle of size $|V|$. The argument presented at the beginning of this section implies that $T_{|V|} \leq 2(|V|-1)$ (Figure 2(b)). It can also be shown that $T_{|V|} \geq 2(|V|-1)$ (see Appendix). Therefore, there is at least one node, v, that transmits at least $2(|V|-1)/|V| = 2-(2/|V|)$ contents (when $|V|=3$, Figure 2(c) illustrates a scheme under which all nodes transmit exactly 4/3 contents). As $|V| \rightarrow \infty$ the number of transmissions from v tends to 2, which proves the tightness of our bound.

III. SYSTEM DESIGN

Our purpose in this and the upcoming sections is to study the impact of design choices, such as how peers are matched, on the efficiency of the system. Our key metric to measure efficiency is the number of transmissions executed by each user before leaving the network. We first assume that users have only local information to make their decisions, and later consider brokers to facilitate exchanges.

A. Overview

We study a time slotted system. At every time slot, users are paired and content exchanges happen between paired users. Each user has capacity to send one content per time slot. A content may correspond to a file or a chunk. If file F is chopped into a set S_F of n_F chunks, $S_F = \{c_{F,1}, \ldots, c_{F,n_F}\}$, users interested in F, when arriving, demand all contents in S_F. For the purposes of the following two sections in most cases it suffices to let $n_F = 1$ for all files. In Section IV-B we discuss a scenario requiring $n_F > 1$.

All users follow a direct reciprocity tit-for-tat strategy, according to which one content is sent only if another one, not yet owned, is received. We consider networks where all members are of the same type, either selective, semi selective or non selective (Figure **??**). An exchange between selective users must involve the transmission of demanded content in both directions whereas non selective users exchange content even if what they receive was not demanded. An exchange between semi selective users occurs if at least one of the users in the pair receives demanded content.

We now relate semi selective exchanges to our main result on the loss of efficiency due to direct reciprocity being two. Let the system loss of efficiency, $\overline{L}$, be defined as the ratio of the aggregate number of transmissions in the direct and indirect reciprocity schedules, $\overline{L} = (\sum_{v \in V} t_v^d)/(\sum_{v \in V} t_v^i)$. Starting from a workload that can be satisfied using indirect reciprocity, $\sum_{v \in V} t_v^i = \sum_{v \in V} |D_v|$. Assuming that under direct reciprocity all users adopt semi selective exchanges, for every two transmissions from paired nodes at least one necessarily delivers demanded content, $\sum_{v \in V} |D_v| \geq \sum_{v \in V} t_v^d/2$. Therefore, $\overline{L} \leq 2$. Note however, that due to the lack of a central controller some users may still transmit more than twice the number of contents than they would in an indirect reciprocity scheme (i.e., the loss of efficiency of a given user, L, may be greater than two) and/or the system may deadlock before all users receive all the desired contents. One of our goals in the following section is to identify scenarios under which semi strict exchanges suffice to allow almost all users to have their demands satisfied, even when users have incomplete information about the system.

B. Workloads

We consider two types of workloads in our experiments, cycles and Zipfian workloads.

1) Cycle Workloads: In a cycle with $|V|$ users and non-overlapping demands, each user v owns content c_v and demands content c_{v-1}, where subtraction is done modulo $|V|$. Unless otherwise stated, the number of contents is $|C| = |V| = 200$ and the number of contents initially owned and demanded by each user is $|C_v| = |D_v| = 1$, $0 \leq v \leq |V|-1$.

In a cycle with overlapping demands, multiple users may request the same content and some users, referred to as contentless, arrive to the network without bringing any contents. In particular, we assume that user v demands content $c_{\lceil v/5 \rceil - 1 \bmod \lceil V/5 \rceil}$ and, if $v \bmod 5 = 0$, owns content $c_{v/5}$. The number of contents in the network is $|C| = |V|/5 = 40$. In a network supporting indirect reciprocity, contentless users may work as relays. In a network enforcing direct reciprocity, one needs to build mechanisms to bootstrap such users, as described in Section IV-B.

2) Zipfian Workloads: In a Zipfian workload, each user demands content k with probability $p_k = 1/k$, and owns content $C-k-1$ with probability p_k, $k = 0, \ldots, C-1$. Note that the most demanded content is also the scarcest one. Unless otherwise stated, we set the number of contents $|C| = 100$, number of users $|V| = 200$ and endowment $|C_v| = 10$, $0 \leq u \leq |V|-1$.

C. Brokers: the Public Board and the Matchmaker

While analyzing each workload we begin by assuming that users are randomly paired and if multiple contents are available for exchange ties are broken arbitrarily. We then progressively add complexity to the system, by letting users to decide which contents to send based on a public board and allowing a matchmaker to decide how users are paired, as described next.

1) The Public Board: When users have multiple contents to offer, a public board may help them decide which content to transmit. We consider a public board that informs, for each content, at each time slot, the number of users that own

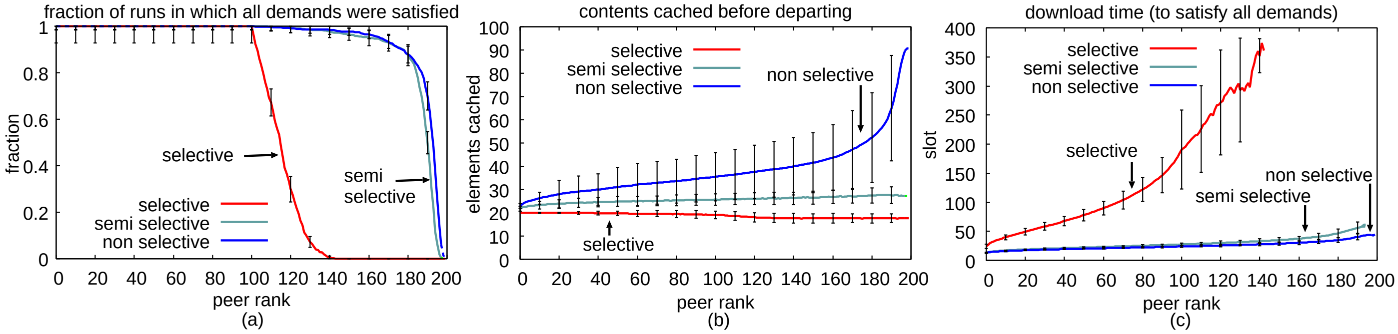

Fig. 5. Zipfian workload. Users benefit from prefetched content and semi selective exchanges present a good compromise between performance and overhead: (a) fraction of users that satisfied all demands; (b) elements cached when leaving the network (equals transmissions plus endowment); (c) download times.

and demand it. If a user has multiple contents to offer to its neighbor, the candidate with highest ratio of number of copies demanded over number of replicas available is selected. Conversely, if a user needs to evict a content from its cache, the one with the lowest ratio is chosen.

2) The Matchmaker: A matchmaker may pair users based on the contents that they own and demand. In this paper, the matchmaker uses a maximum weight matching algorithm to pair users with weights set as follows. Users are randomly divided into two subsets V_1 and V_2 of equal size (if the network has an odd number of users, one user is randomly discarded). There is an edge from a user $v_1 \in V_1$ to $v_2 \in V_2$ with weight 2, 1 and $\epsilon = 0.1$ if v_1 and v_2 can establish a selective, semi selective and non selective exchange, respectively. There is no edge between users that cannot exchange contents. Prioritizing matches between users that can exchange contents, the matchmaker increases the throughput of the system, measured by the number of useful exchanges per time slot.

IV. BARTERING AND BROKERS

Our goal now is to illustrate *(a)* the system benefits of users prefetching contents even when the contents are not of their immediate interest and *(b)* the impact of brokers on the performance of users. For this purpose, we analyze both the cycle and the Zipfian workloads.

A. Bartering: The Benefits of Content Prefetching

We now evaluate the implications of users downloading non demanded contents, for bartering purposes. In the cycle workload, bartering is essential. If users are selective, no exchanges can occur and the system remains deadlocked forever. On the other hand, our experiments indicate that if users are not selective they can have their demands satisfied in an average of 83.63 time slots, even without a broker.

In a Zipfian workload, bartering decreases the chances that a content becomes unavailable before all requestors are able to download it, simply because more contents get replicated if users take advantage of their exchange opportunities. Note, however, that in the face of lack of information about which content to barter or which user to contact, bartering may lead to an increase in the number of transmissions without a significant increase in the availability of the contents or in the download times.

To illustrate the observations made above, we consider a Zipfian workload where the number of contents initially demanded and owned by each user is 10 (last column of Table II), users don't have constraints on their cache sizes and on the time they spend in the system, and the public board is available. We repeat our experiments 1000 times, each experiment ending when either all users satisfy their demands or a deadlock is reached. Figure 5 shows (a) the fraction of users that were able to complete the download, (b) the number of contents in the cache when the user left the network and (c) the time that each user took to download all the demanded contents. The x-axis in all the three figures is the ranking of a user, e.g., the 10^{th} user to leave the system was always able to have its demand satisfied while the 200^{th} was never able to download all the requested contents (Figure 5(a)). Note that some curves in Figures 5(b) and 5(c) are not monotonically increasing because the metrics showed are conditioned on the peers concluding their downloads. The higher the peer rank, the smaller are the chances that the peer satisfies all its demands and the smaller the number of samples collected.

The performance of non selective and semi selective users is better than the performance of the selective ones. The fraction of completed downloads is higher and the download times are smaller when users are non selective or semi selective as opposed to selective. To achieve high performance, the overhead incurred by non selective users is the additional number of contents cached before leaving the system. Semi selective users present the best outcome, with high performance (small download times and large number of conclusions) and low overhead (small number of contents cached before leaving the system).

Even though with selective exchanges a significant fraction

	workload			
demand/ownership graph	cycle	cycle	Zipfian	Zipfian
users leave network?	no	no	yes	yes
contentless users	no	yes	no	no
overlapping demands	non overlapping	overlapping	overlapping	overlapping
demand per user	1	1	2	10
exchange strategy	outcome			
selective	*generally unfeasible*			
with matchmaker	deadlock	deadlock	deadlock after few iterations	loss of eff. close to 0
without matchmaker	deadlock	deadlock	deadlock after few iterations	frequent deadlocks
semi selective	*generally feasible, incurring small loss of efficiency*			
with matchmaker	user loss of eff. ≤ 2	user loss of eff. for sources ≤ 2	system loss of eff. ≤ 2 frequent deadlocks	loss of eff. close to 0 very unfrequent deadlocks
without matchmaker	system loss of eff. ≤ 2	system loss of eff. for sources ≤ 2	system loss of eff. ≤ 2 frequent deadlocks	system loss of eff. ≤ 2 very unfrequent deadlocks
non selective	*generally feasible, incurring large loss of efficiency*			
with matchmaker	user loss of eff. ≤ 2	loss of eff. for sources ≤ 2	loss of eff. ≥ 2	loss of eff. close to 0
without matchmaker	loss of eff. ≥ 2	loss of eff. for sources ≥ 2	loss of eff. ≥ 2	loss of eff. ≥ 2
optimistic bootstrap	nonessential, except for selective exchanges	essential if all users are to receive all demanded contents	helpful to avoid deadlocks for sel. and semi sel. exchanges	helpful to avoid deadlocks for sel. exchanges

TABLE II
DIRECT RECIPROCITY DESIGN SPACE: EXCHANGE STRATEGIES, WORKLOADS AND RESPECTIVE OUTCOMES.

of users are not able to have all their demands satisfied, most users download a considerable amount of the requested contents. We observed that accross our simulations 90% of selective and semi selective users were able to conclude all their downloads. This fraction drops to 50% for selective users. Nevertheless, even with selective exchanges more than 90% of the users were able to download at least eight out of the ten demanded contents (see [12]).

B. Brokers: The Role of Information on the Loss of Efficiency

We now analyze the role of brokers to recommend contents to users (public board) and pair them efficiently (matchmaker). As in the previous section, we begin by analyzing the simple cycle workload. In cycles with semi selective exchanges, the system loss of efficiency, $\overline{L}$, is at most two. However, without a matchmaker some users may need to transmit more than two contents to their neighbors. This happens, for instance, if in the first three time slots user v_2, with endowment c_2, is matched with v_3, v_4 and v_1, in that order, transmitting three contents, c_2, c_3 and c_1, respectively. In contrast, with a matchmaker each user transmits at most two contents. In the first (resp., second) time slot users with even (resp., odd) index transmit useful content to users with odd (resp., even) index, and the loss of efficiency incurred by each user is exactly two. Finally, if users are not selective, the system loss of efficiency may be higher than two, which happens, for instance, if in the first time slot user $v_{i \bmod V}$ is matched with user $v_{i+2 \bmod V}$, $0 \leq i \leq |V|-1$.

If the cycle involves contentless users (second column of Table II) the observations made in the paragraph above are still applicable for the network involving only the sources. However, if all users are to receive all demanded contents, a mechanism to bootstrap the contentless users is essential. One solution consists in giving them credits. Those credits can then be swapped for other contents.

A second solution, more in line with optimistic unchoke in BitTorrent, consists in allowing some users to send a small number of chunks to their contentless neighbors, without being immediately reciprocated. If the contentless neighbors are able to complete their downloads only after offering chunks back to the original donor (such schedule being enforced by the matchmaker), direct reciprocity will still hold.

In Zipfian networks brokers are particularly important when users have limited cache sizes and delay constraints. Consider a system in which each user can cache 10 files and leaves the network either when it has satisfied its demand or after 40 time slots have been elapsed, whatever comes first. As soon as a user leaves the system, it is replaced by a new one. The other parameters are those described in the last column of Table II.

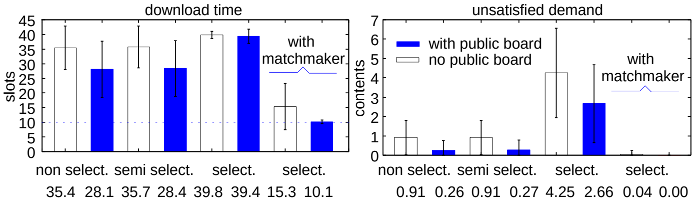

Fig. 6. Benefits of a public board and a matchmaker with a Zipfian workload are significant. Maximum residence time of users is 40 and cache size 10.

Figure 6(a) shows the download times experienced by the users with non selective, semi selective, and selective exchanges. The more selective are the users, the larger the delays they incur. Figure 6(b) shows the fraction of unsatisfied demand, which also increases as users become more selective. As one might expect, the use of the public board, both to decide which content to send as well as to decide which content to evict, plays a key role in this scenario. The public board reduces the delay as well as the fraction of unsatisfied demand, as can be observed comparing the blue and white bars, corresponding to the system with and without the public board.

More surprising is the effect of the matchmaker (bars on the right of Figure 6). The use of a matchmaker coupled with a public board yields an optimal delay of 10 time slots with all users having their demands satisfied. The rationale behind this outcome goes as follows. While the public board helps unpopular contents to be replicated early on, the matchmaker guarantees that all users are matched in such a way that every time slot is used to transmit demanded contents.

The scenario above drastically changes if users request only 2 contents when joining the network (third column of Table II). In this case, even with brokers semi selective exchanges no longer suffice to guarantee that all users get all demanded contents. Non selective exchanges must be employed, and the system loss of efficiency may be greater than two.

V. RELATED WORK

The literature on incentives in peer-to-peer systems focusing on its relationship to clustering [13], freeriding [14] and system design [15] is vast. Nevertheless, we were not able to find any previous study on the *fundamental* pros and cons of direct and indirect reciprocity for the dissemination of digital goods.

In the economics literature, direct reciprocity bartering schemes are considered precursors of more sophisticated economies involving money and indirect reciprocity [10]. Arpejis et al. [7] argue that prices can also play an important role in peer-to-peer systems. The authors compare implicit prices in bilateral exchanges (direct reciprocity) with explicit prices in multilateral exchanges (indirect reciprocity) and propose a system that implements the later. Our work, in contrast, suggests that in the context of peer-to-peer file sharing, direct reciprocity has its own advantages and might suffice to implement an efficient and incentive-compatible system.

In the economics terminology, identifying an indirect reciprocity schedule is referred to as market clearing [9]. In this paper, we assumed that an indirect reciprocity schedule is provided. Future work consists in studying how to efficiently find such schedule, noting that to find indirect reciprocity cycles in a distributed fashion one can use distributed deadlock detection algorithms [16].

Our work is related to those on scheduling in single swarm peer-to-peer systems [17]. Reciprocity constraints were first incorporated in peer to peer schedules by Figueiredo et al. [18]. While [18] focus on cooperative versus non cooperative systems, we focus on direct versus indirect reciprocity. In addition, we address multi-commodity scheduling, which has received little attention in the realm of peer-to-peer systems (one exception being [19]).

VI. CONCLUSION

Direct reciprocity is a key of BitTorrent's success, one of the most popular peer-to-peer systems nowadays. In this work we provide new foundational results on reciprocity mechanisms. In the context of our proposed model, we show that the loss of efficiency due to direct reciprocity is at most two. Then, using simulations, we indicate that in many situations direct reciprocity can lead to high performance with marginal overhead. In particular, we identified that in cycles and Zipfian workloads it is crucial that users download contents for the purpose of bartering, and a low loss of efficiency may be achieved if brokers are available to perform matchmaking between users and to issue recommendations on content value. Although our model is based on a number of simplifying assumptions, such as all contents having the same size and all peers following the same set of rules, we believe that it sheds important insights on the fundamental limitations and potentials of direct reciprocity.

Acknowledgement: Part of this work was done while DT and DSM were visiting Thomson Research Labs. This work was supported in part by the NSF under award numbers CNS-0519922 and CNS-0721779. Research of DSM funded in part by a scholarship from CAPES/Fulbright (Brazil).

REFERENCES

[1] "Titletrader, swap your stuff," http://www.titletrader.com/.

[2] D. S. Menasche, A. A. Rocha, B. Li, D. Towsley, and A. Venkataramani, "Content availability and bundling in swarming systems," in *CONEXT*, 2009.

[3] M. A. Nowak and K. Sigmund, "Evolution of indirect reciprocity," *Nature*, vol. 437, pp. 1291–1298, 2005.

[4] D. Levin, K. LaCurts, N. Spring, and B. Bhattacharjee, "Bittorrent is an auction," in *SIGCOMM*, 2008.

[5] M. Piatek, T. Isdal, T. Anderson, A. Krishnamurthy, and A. Venkataramani, "Do incentives build robustness in bittorrent?" in *NSDI*, 2007.

[6] "eMule project," http://www.emule-project.net/.

[7] C. Aperjis, M. J. Freedman, and R. Johari, "Peer assisted content distribution with prices," in *CONEXT*, 2008.

[8] L. Xiong and L. Liu, "Peertrust: Supporting reputation-based trust for p2p electronic communities," *IEEE Tran. Knowl. Data Eng.*, 2004.

[9] D. J. Abraham, A. Blum, and T. Sandholm, "Clearing algorithms for barter exchange markets: Enabling nationwide kidney exchanges," in *ACM Conference on Electronic Commerce*, 2007.

[10] R. Lagos and R. Wright, "When is money essential?" in *http://www.ssc.upenn.edu/~rwright*, 2007.

[11] B. Bollobas, *Modern Graph Theory*. Springer Verlag, 1998.

[12] L. Massoulie, D. S. Menasche, and D. Towsley, "Reciprocity in bartering systems," in *University of Massachusetts Technical Report*, 2009.

[13] A. Legout, N. Liogkas, E. Kohler, and L. Zhang, "Clustering and sharing incentives in bittorrent systems," in *SIGMETRICS*, 2007.

[14] T. Locher, P. Moor, S. Schmid, and R. Wattenhofer, "Freeriding in bittorrent is cheap," in *HotNets*, 2006, pp. 85–91.

[15] M. Piatek, T. Isdal, A. Krishnamurthy, and T. Anderson, “One hop reputations for peer to peer file sharing workloads,” in *NSDI*, 2008.

[16] K. M. Chandy, J. Misra, and L. M. Haas, “Distributed deadlock detection,” *Transactions on Computer Systems*, no. 2, pp. 144–156, 1983.

[17] J. Mundinger, R. Weber, and G. Weiss, “Optimal scheduling of peer-to-peer file dissemination,” *Journal of Scheduling*, 2008.

[18] D. D. Figueiredo, B. Venkatachalam, and S. Wu, “Bounds on the performance of p2p networks using tit-for-tat strategies,” in *IEEE Conference on P2P Systems*, 2007, pp. 11–18.

[19] L. Massoulie and A. Twigg, “Rate-optimal schemes for peer-to-peer live streaming,” *Perform. Eval.*, vol. 65(11-12), pp. 804–822, 2008.