A Sybilproof Indirect Reciprocity Mechanism for Peer-to-Peer Networks

Raul Landa, David Griffin, Richard G. Clegg, Eleni Mykoniati and Miguel Rio
Department of Electronic and Electrical Engineering, University College London
Email: {rlanda, dgriffin, rclegg, emykoniati, mrio}@ee.ucl.ac.uk

***Abstract*— Although direct reciprocity (*Tit-for-Tat*) contribution systems have been successful in reducing freeloading in peer-to-peer overlays, it has been shown that, unless the contribution network is dense, they tend to be slow (or may even fail) to converge [1]. On the other hand, current indirect reciprocity mechanisms based on reputation systems tend to be susceptible to *sybil attacks*, *peer slander* and *whitewashing*.**

In this paper we present PledgeRoute, an accounting mechanism for peer contributions that is based on *social capital*. This mechanism allows peers to contribute resources to one set of peers and use this contribution to obtain services from a different set of peers, at a different time. PledgeRoute is completely decentralised, can be implemented in both structured and unstructured peer-to-peer systems, and it is resistant to the three kinds of attacks mentioned above.

To achieve this, we model contribution transitivity as a routing problem in the *contribution network* of the peer-to-peer overlay, and we present arguments for the routing behaviour and the sybilproofness of our contribution transfer procedures on this basis. Additionally, we present mechanisms for the seeding of the contribution network, and a combination of incentive mechanisms and reciprocation policies that motivate peers to adhere to the protocol and maximise their service contributions to the overlay.

I. Introduction

Peer-to-peer overlays coordinate the contributions of large numbers of independent peers to form scalable, decentralised, self-organising content delivery systems. These are, however, susceptible to the *freeloading problem* [2], [3]: it is individually rational for each peer to contribute as little as possible, while at the same time consuming the contributions of other peers.

If resource contribution is framed as an *Iterated Prisoner's Dilemma*, the expectation for reciprocity in future interactions (including retaliation) can be a strong incentive for cooperation [4]. Reciprocity thus emerges as a viable technique for the control of freeloading. We distinguish two different kinds of reciprocity:

- **Direct Reciprocity**, where the interaction between two peers is only influenced by past interactions between them.
- **Indirect Reciprocity**, where peer interactions are not only influenced by their mutual past interactions, but by their interactions with other peers as well.

Many resource allocation techniques based on direct reciprocity have been proposed (see, for instance, [5]–[7]). These, although easily implementable on the peers, can be limited by the high churn rates typical of peer-to-peer overlays. Any given peer may be interacting essentially with strangers, and thus, direct reciprocity mechanisms will tend to be slow (or may even fail) to converge [1]. In practice, these mechanisms are only appropriate for systems with long-lived relationships where there is ample opportunity for mutual reciprocation.

Direct reciprocation is further complicated if the subset of peers that provide the services that a peer seeks is not the same one that is interested on the services it provides. This supply-demand mismatch can also happen in time, since peers are usually only interested in particular services at particular times. Thus, a peer can find that its past contributions are useless when it comes to obtaining services in the present. In these circumstances, a reciprocity system that allows contributions given to a peer to be repaid by other peers at different times in the future may be the only way to foster cooperation between strategic peers seeking private utility maximisation.

Most indirect reciprocity schemes proposed in the literature (see, for instance [6]–[8]) are based on *reputation systems*, and differ mainly in the way the reputation scores are calculated and propagated. In general, however, they assign ratings to peers according to their past behaviour, and communicate them through the overlay. This makes them vulnerable to exploitation, due to the near-zero cost of creating new identities. Peers might create arbitrary contributions between fictitious peers (the *sybil attack* [9]), lie regarding their contributions or the contributions of other peers (the *slander* attack), or discard identities that have been libelled as malicious and penalised (the *whitewashing* attack).

Other indirect reciprocity schemes rely on implementing a currency-based economy that is resistant to forgery and double-spending (see, for instance, [10] and [11]). A common problem of these schemes is the minting of currency and the trust anchoring that it implies, usually necessitating either a public key infrastructure, a web of trust, or threshold cryptography techniques. Furthermore, these systems usually make use of auctions for price setting, which might slow their convergence.

In this paper we propose **PledgeRoute**, a contribution accounting system that allows peers to contribute to a set of peers and transfer these contributions for use with other peers in a direct reciprocity basis. Thus, we enable indirect reciprocity by decomposing it in two stages: *contribution transfer* and *direct reciprocation*.

To achieve contribution transfer, we propose an accounting

system that not only keeps track of peer contributions, but also allows them to be transferred between peers following transitive contribution chains. Thus, peers can contribute to one set of peers and transfer these contributions to another set of peers from which services are required, implementing a distributed exchange economy.

By casting contribution transitivity as a routing problem, we can use distributed routing algorithms to achieve our objective in a completely decentralised fashion. Thus, we propose a Dijkstra-inspired generalised routing algorithm to route contributions through the network by means of *wealth-preserving* transactions. In order to make the system sybilproof [12], our algorithm for contribution transfer relies on finding the end-to-end transfer paths that have maximum bottleneck contributions, which have sybilproofness properties similar to those of maximum flow. Thus, since contributions are always bound to an identity, peers gain nothing from having multiple identities: their contributions will be simply split among them. To complement our routing algorithm, we present simple cryptographic techniques and protocol operations that provide resistance to slander and whitewashing attacks.

To avoid the execution of costly routing computations over the entire peer-to-peer overlay topology, we propose a probabilistic topology sampling algorithm based on a truncated, self-avoiding random walk that samples preferentially those paths capable of yielding high-valued contribution transfers. Contribution transfer is achieved using a soft-state reservation protocol.

Since the contribution transfer operation only preserves the net contributions of each peer, and not its absolute contributions and obligations, there is a danger of draining the contribution network of social capital (we discuss this problem in Section V). Thus, although no peer ends up "worse off", the capacity of the system to transfer contributions from one peer to another will be reduced. To counter this, we propose a technique to seed the contribution network based on the creation of a credit tree rooted on each peer, and its use for the creation of balanced contribution cycles.

To achieve stable cooperation in the reciprocation phase in the presence of strategic peers, we propose an incentive mechanism based on the modification of the contribution values of each peer.

Our contributions are presented in three main parts. Firstly, we present the generalities of **PledgeRoute** (Section III), our proposed probabilistic topology sampling algorithm (Section IV-A) and our contribution transfer protocol (Section IV-B). Secondly, we present our proposed algorithm to seed the contribution network by identifying trust cycles (Section V). Thirdly, we describe our incentives system (Section VI). In Section VII we present the attack resistance properties of our proposed scheme, and in Section VIII we present some evaluation results.

II. Definitions

In this paper we will be mainly concerned with the *contribution network*: the graph-theoretic representation of the contributions that have been given and received in the peer-to-peer overlay. We will model the contribution network as link-weighted, directed graph $\mathcal{G}$: a finite set of peers $N = \{n_i\}$ and a set of links $L = \{l_{ij}\} \subset N \times N$, where each l_{ij} is associated with a *weight* $w_{ij} \geq 0$ that corresponds to the magnitude of the contributions from n_i to n_j. In practice, w_{ij} is just an account associated with n_i that is maintained in n_j. We will denote a *simple path* in $\mathcal{G}$ from n_i to n_j as P_{ij} (we discuss the weighing of paths and links in the context of our routing algebra in Sections IV and VII).

We will call the set of all peers reachable through outgoing links of n_i its *outgoing peer neighbourhood* $\Pi^-(n_i)$ (its *incoming peer neighbourhood* $\Pi^+(n_i)$ is defined similarly). We shall denote the outdegree of n_i in $\mathcal{G}$ (the total unpaid contributions that n_i has given to its neighbours) as its *social capital*, and define it as $C(n_i) = \sum_{j \in \Pi^-(n_i)} w_{ij}$. Conversely, we shall denote its indegree in $\mathcal{G}$ (the total unpaid contributions that n_i has received from its neighbours) as its *pledged resources*, and define it as $R(n_i) = \sum_{j \in \Pi^+(n_i)} w_{ji}$. We call the net contribution of n_i to the system its *wealth*, and we define it as $W(n_i) = C(n_i) - R(n_i)$.

We shall be interested in modifying the topology of $\mathcal{G}$ (usually by creating new links) while maintaining the net contributions of each peer invariant. To perform such *wealth preserving* transformations, we begin by defining a closed cycle $\Sigma \in \mathcal{G}$ on the contribution network, and we assign an orientation to it. If it is necessary to create new links in $\mathcal{G}$, we consider them to have zero weight ($w_{ij} = 0$).

Having defined Σ, we traverse its links following its orientation, adding an amount τ to the contributions associated with each link that is traversed in its designated direction, and subtracting an amount τ to the contributions associated with each link that is traversed opposite to its designated direction (transformations requiring negative link weights are considered invalid). Evidently, each peer in Σ adds the same amount to its $C(n_i)$ and its $R(n_i)$, thus keeping its wealth invariant. After the transformation, links with zero weight are removed from $\mathcal{G}$.

Every peer n_i sets u_{ij}, the maximum amount of unreciprocated contributions that it will give to a neighbour n_j in $\mathcal{G}$. We will call u_{ij} the *safe credit margin* of n_j as assessed by n_i. In order to calculate u_{ij}, each peer n_i maintains $\hat{w}_{ij}$, the amount of contributions that n_j has given to n_i as response to reciprocative requests (see Section III). As peers successfully reciprocate for a number of interactions they can be more reliant on the continued trustworthiness of their neighbours, and u_{ij} will increase with $\hat{w}_{ij}$. Finally, each peer enforces ξ_i, the maximum credit it is willing to support. We defer to Section V the analysis of the relation between w_{ij}, $\hat{w}_{ij}$ and u_{ij}.

III. Basic System Architecture

In our system, the contribution network $\mathcal{G}$ represents the set of previously rendered services, which carries a reciprocity implication and is thus equivalent to the notion of *social capital* in the social sciences [13].

We require no centralised identity management system. Instead, peers use self-certifying identifiers that are exchanged when they initially come into contact. These can be used as public keys, to verify digital signatures on the messages sent by their neighbours. All communication between peers is digitally signed.

There are two distinct interaction policies in our system: *altruistic* and *reciprocating*. A peer chooses one of these two when requesting a service, and the behaviour of the server peer depends on this choice.

When a peer requests a service using the altruistic policy, it does not offer any kind of payment for it, relying instead on the altruism of the serving peer. The serving peer, however, will only grant the service in a best-effort basis: its service quality will be reduced as required, giving priority to reciprocation-based interactions.

However, altruistic services are not free: the serving peer n_j will expect reciprocation at some future time from the requesting peer n_i (this is equivalent to the *reciprocal altruism* of [14]). This expectation takes shape as an increase in w_{ji}, and models the reciprocative obligation that the recipient has contracted towards the serving peer.

Although any request using the altruistic policy receives essentially random service quality, this policy is of fundamental importance as it helps the system *bootstrap*: when new peers arrive to the system, they have no previous contributions to other peers in the overlay, and they can only obtain services through the altruism of other peers (at least until they have the opportunity to provide services themselves).

When a peer n_i requests a service from a server peer n_j using the reciprocating policy, it will offer a payment τ to cover it that will be deducted from the account w_{ij} in n_j (the previous contributions that n_i has made to n_j, including contributions that n_i has transferred to n_j from other peers). If τ is sufficient to cover for the request, n_j will give it prioritised service and subtract τ from w_{ij}. If τ is insufficient or $w_{ij} < \tau$, the request will be interpreted as an altruistic interaction, and the request originator will be informed of this fact. This should be uncommon, as the originator is aware of its contributions to the server peer. We assume that the conversion between services and contribution units is the same for all peers, and universally known.

Clearly, there is an element of trust involved in this transaction: the value of the service that the client peer receives might not be equivalent to the amount τ by which its account is decreased. This problem is equivalent to the server n_j arbitrarily decreasing the client's account w_{ij} by an amount equal to the difference between τ and the true value of its service[1]. This is further explored in Section VI.

In summary, thus, altruistic interactions **increase** the social capital in the network, while reciprocative interactions **decrease** it. We now explore the use of these accounts for contribution transfer.

[1]The problem of the client refusing to pay the server after it has granted a service does not exist, as the server locally maintains the account with the contributions of the client.

IV. Distributed Contribution Accounting

A. Contribution Topology Discovery

The first issue that we will address is the process that allows each peer to construct a local view of the $\mathcal{G}$ that can be used for the computation of contribution transfers. This local view of $\mathcal{G}$ from the standpoint of n_i will be called $\mathcal{G}_i$. The aim of our topology discovery algorithm is to allow peers to attain high levels of contribution transfer capability while keeping as few links of $\mathcal{G}$ in $\mathcal{G}_i$ as possible.

The contribution topology discovery process starts when a peer advertises the unpaid contributions that it has to each of its neighbours in $\mathcal{G}$ using *Pledge Announcement Messages* (PAMs). Each PAM is associated with a simple path in $\mathcal{G}$: a series of links terminating on the sending peer where no peer or link is repeated.

The path that the PAM follows is decided in a probabilistic fashion, with the probability of it being forwarded from a peer n_i to a given neighbour peer n_j made proportional to w_{ji}, the unpaid contributions that n_j has made to n_i (the pledged resources that n_i has towards n_j).

We model this using a truncated, self-avoiding random walk defined on the contribution network $\mathcal{G}$. Peers periodically generate PAMs which are sent in random walks following the links in $\mathcal{G}$. These random walks, however, are biased to preferentially traverse links that have high contribution values, yielding biased topology samples that favour subgraphs with high contribution transfer potential.

We focus on the transition probability associated with a single instance of the random walk starting in peer n_s. If we define $p_j^t(s)$ as the probability of a PAM starting at peer n_s visiting peer n_j at time t, the state transition probability for our random walk is given by

$$p_j^{t+1}(s) = \rho_{\text{stop}}\delta_{js} + (1-\rho_{\text{stop}}) \sum_{i \in \Pi^+(n_j)} r_{ij}^t p_i^t(s) \ , \qquad (1)$$

where r_{ij}^t denotes the probability of a transition from state i to state j (the forwarding of the PAM from n_i to n_j) at time t, ρ_{stop} denotes the probability that the PAM walk is terminated, and δ_{js} is the Kronecker delta. It is apparent from (1) that the system will periodically regress to its initial state s, corresponding to peer n_s. This reflects the fact that the PAM will be removed from the network, and eventually a new one will be generated by the originating peer n_s.

In order to make (1) self-avoiding and biased towards high-contribution paths, we define for the transition probabilities that

$$r_{ij}^t = \begin{cases} \frac{w_{ji}}{\sum_{j \in U(n_i,P)} w_{ji}} & \text{if } U(n_i,P) \neq \emptyset \\ \delta_{js} & \text{otherwise.} \end{cases} \qquad (2)$$

In (2), $U(n_i, P)$ is the *unvisited neighbour set* of n_i for path P: $U(n_i,P) = \Pi^-(n_i) \backslash P$. Informally, $U(n_i,P)$ is the set of peers that have unreciprocated contributions towards n_i which have not forwarded the PAM yet. Thus, r_{ij}^t will favour transitions from n_i to peers n_j with large w_{ji} values, avoiding previously visited peers, and terminating the walk if

$U(n_i, P) = \emptyset$. In practice, this random walk is implemented by including the sequence of visited peers on each PAM. Every peer will extract the neighbours that have not been visited by the PAM and then pick one randomly, the distribution being weighted by the contributions that each eligible neighbour has given to n_i (the value of the w_{ji} account). Thus, peers that can support higher contribution transfers are preferentially selected. After a given peer (say n_j) is selected for forwarding, n_i will insert in the PAM a timestamp, a nonce, the self-certifying identifier of n_j, the maximum contribution value w_{ji} available for contribution transfer, and will digitally sign of all these. When n_j receives the PAM, it can check that it was indeed originated by n_i, that it is actually directed to itself, that the contribution value w_{ji} reported by n_i is correct, and that it is not a replayed message. This last check is done by ignoring all expired PAMs (those with timestamps older than a given timeout value t_{old}), and keeping track of all the nonces corresponding to messages that have not expired.

As the PAM propagates according to (1) and (2), each peer will include its own identifier, the contribution value that it has for the next peer in the chain and the rest of the elements detailed above. Given that every peer can trivially verify the w_{ji} claimed by an upstream PAM peer, every PAM message is audited at each step of the random walk and it can be considered truthful.

After processing the PAM, every peer n_i on P will be able to update its local contribution subgraph $\mathcal{G}_i$ with the w_{ij} of the links that the PAM traversed before it reached n_i. Since PAMs are forwarded following links in $\mathcal{G}$ end-to-start, each peer will update its local view of the contribution network with a directed contribution path rooted on itself and terminating on the originator of the PAM. When a given peer decides to terminate the PAM walk, it will forward it back to its originating peer (the reason for this is explored in Section VI).

B. Contribution Transfer Protocol

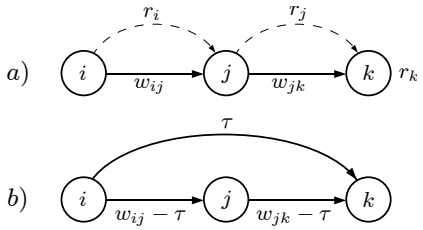

Fig. 1. Contribution paths and transfers

Once a subgraph $\mathcal{G}_i$ of $\mathcal{G}$ has been discovered by incorporating the information of a number of PAMs, the next step is to use it to perform contribution transfers (See Fig. 1).

Contribution transfer events can be decomposed as consecutive pairwise transactions. Thus, past contributions are only needed between adjacent peers in the contribution chain. However, since $\mathcal{G}_i$ may not perfectly reflect $\mathcal{G}$, we design the transfer protocol to be resistant to errors in $\mathcal{G}_i$.

If n_i requests to transfer up to r_i contribution units to a "remote" peer n_k, the next-hop peer n_j might be unable to immediately commit, as this is contingent on changing conditions in the downstream contribution chain. Instead, n_j can calculate a contribution value that itself is willing to reserve for the transaction (r_j in Fig. 1) and forward this information on. If every peer in the contribution chain does this, the final peer in the chain (in this case n_k, the peer that will actually provide the service) will have a complete view of the amounts that each peer is willing to commit (including itself), and will be able to compute an amount τ that complies with all these requirements (Fig. 1a)[2].

After τ is found, it is deducted from the accounts w_{ij} in the contribution path, and added to the account for the origin peer on the destination peer (w_{ik} in Fig. 1b). Thus, the final result of a contribution transfer of magnitude τ is to create a contribution link of τ from n_i to n_k through a wealth preserving operation, subtracting τ from every hop in the contribution chain to compensate for the creation of the new link.

PledgeRoute implements transfers using *Contribution Transfer Request Messages* (CTRMs) that are source routed to their destination peer n_k according to $\mathcal{G}_i$. CTRMs are structurally very similar to PAMs, including the unreciprocated contributions w_{ij}, the reserved contributions r_j and the self-certifying identifiers of all peers that they traverse, as well as their nested digital signatures. Each CTRM traverses the transfer path twice: once from n_i to n_k, setting up the transfer reservations, and once in the reverse direction, when the link from n_i to n_k is formed and τ is subtracted from the others.

As is common with soft-state reservation protocols, the reserved contributions on each one of the peers in the contribution chain will be automatically released if the end-to-end transaction is not successfully completed within a timeout interval t_{T}. In case of protocol failure, a failure message propagates back along the contribution chain, in order to free the reserved contribution amounts in previous peers without having to wait for t_{T} to elapse.

Once contribution has been transferred, it is indistinguishable from direct contributions, and thus peers are able to flexibly decide where to spend the accumulated pledged resources that other peers maintain for them.

C. Contribution Transfers as a Routing Problem

By using the routing algebra model of [15], we can analyse the contribution transfer operations formally. In particular, we see that the routing of CTRMs according to maximum transfer capacity paths can be modelled with a routing algebra where $\min$ is applied over the edges of paths to calculate their capacity, and $\max$ is used to compare the desirability of different paths, preferring paths with greater capacities. Formally, our system can be modelled as the *max-min semiring* over the real numbers, $(\mathbb{R}, \max, \min)$, which is normally used to model bandwidth-aware routing.

[2]Of course, $\tau \leq \min(w_{ij}, w_{jk})$ and the maximum τ feasible on the contribution network coincides with the maximum flow over the simple path along the unreciprocated contributions chain, which makes it sybilproof (see Section VII).

A useful property of this algebra is that it is *monotonic*: the addition of links to paths preserves the ordering of paths in terms of their desirability. This means that if we have two paths P_{ij} and P_{ik} starting from n_i, and P_{ij} is weakly preferred over P_{ik} (denoted as $P_{ij} \succsim P_{ik}$), then prepending a new path P_{si} to both P_{ij} and P_{ik} to form two alternative paths $P_{sj} = P_{si} \cup P_{ij}$ and $P_{sk} = P_{si} \cup P_{ik}$ will preserve the ordering (desirability) of their path transfer capacities, and we have that $P_{sj} \succsim P_{sk}$. This allows peers to find optimal contribution transfer paths incrementally, by advertising all incoming links and their weights w_{jk} to all its neighbours in $\mathcal{G}$, and running a Dijkstra-like algorithm [16] to route over the contribution network $\mathcal{G}$. In our case, this "link flooding" approach is avoided both for increased scalability and to preserve the slander resistance properties of the system (see Section VII), but each peer n_i uses a Dijkstra-based algorithm locally over its sampled topology $\mathcal{G}_i$ to determine transfer paths.

V. Trust and Contribution Cycle Seeding

It is clear that the contribution transfer process detailed in Section IV-B drains the contribution network $\mathcal{G}$, as it subtracts τ from all links in the CTRM path, while only producing a single new one of weight τ between the transaction originator and the last peer in the chain. Although this is not harmful to peers, because they still have the same wealth (their contributions are reduced in the same amount as their pledges), it is deleterious to the capacity of the network to perform contribution transfers.

There are two processes that can be used to counter this effect by injecting social capital into $\mathcal{G}$. The most fundamental one is *true altruism*: when peers give contributions without decreasing any account in $\mathcal{G}$, there is a net increase in social capital. The second process is *mutual crediting*, in which an hypothetical contribution of value zero is decomposed into two nonzero contributions, but with opposite directions. However, just as *Tit-for-Tat* can be generalised to cyclical multiparty exchanges, mutual crediting can be extended to any closed cycle of consistently oriented links in $\mathcal{G}$. None of the peers involved would experience an increase in wealth, but a number of links with nonzero weights would be added to $\mathcal{G}$.

To accomplish this, we present a process by which peers can find sets of other peers that can agree to create new contribution cycles between them. This means that each one of the peers in the set will pledge some resources to a *successor* peer and gain a pledge for the same amount from a *predecessor* peer, in a ring of shared obligations. Since the wealth of every peer in one of these contribution credit cycles remains unchanged, the only effect of this operation is to replenish link contributions and facilitate contribution transfer.

Naturally, each peer must be confident that the next peer in the cycle will actually reciprocate these virtual contributions in case it is requested to do so. Thus, every peer must estimate the amount of contributions that it can be confident that each of its neighbours would eventually pay back with high probability: the *safe credit margin* u_{ij} defined on Section II.

We calculate u_{ij} as a monotonically increasing, sub-linearly growing function of $\hat{w}_{ij}$ (recall from Section II that the $\hat{w}_{ij}$ are the contributions that n_j has reciprocated to n_i). The reason for this is that, in order to make the safe credit margin reliable, peers will demand an increasing amount of successfully reciprocated contributions in order to increase u_{ij}. We present one such function, where the increase in $\hat{w}_{ij}$ required to allow for an increase in u_{ij} levels grows linearly. To present the intuition behind this function, let us consider two new peers n_i and n_j, such that $w_{ij} = w_{ji} = 0$ and $\hat{w}_{ij} = \hat{w}_{ji} = 0$. Since n_i is unable to ascertain the trustworthiness of n_j (and vice-versa), they will only provide each other with a basic amount of altruistic service: $u_{ij} = u_{ji} = \kappa$.

If peer n_i now gives altruistic contributions to peer n_j amounting to κ, then $w_{ij} = u_{ij} = \kappa$, and no more contributions are possible from n_i to n_j until reciprocation takes place. When n_j reciprocates by providing a service worth κ to n_i, n_j will set $w_{ij} = 0$, and n_i will set $\hat{w}_{ij} = \kappa$ and $u_{ij} = 2\kappa$. Increasing u_{ij} from 2κ to 3κ, however, will require a further increase of $\hat{w}_{ij}$ of 2κ. In general, an increase of u_{ij} from $n\kappa$ to $(n+1)\kappa$ will require an increase of $n\kappa$ in $\hat{w}_{ij}$. Thus, $u_{ij} = n\kappa$ will require $\hat{w}_{ij} = \kappa + 2\kappa + 3\kappa + \cdots + (n-1)\kappa$, and we have that the safe credit margin u_{ij} can be calculated as

$$u_{ij} = \min\left(\frac{1}{2}\left(\kappa + \left(\kappa^2 + 8\kappa\hat{w}_{ij}\right)^{\frac{1}{2}}\right), \xi_i\right), \tag{3}$$

where the first term inside the min is the positive root of $\hat{w}_{ij} = \frac{(u_{ij}-\kappa)u_{ij}}{2\kappa}$ (by the summation above), and ξ_i is the maximum credit allowed by n_i, as defined in Section II.

Each peer then advertises their u_{ij} values to their neighbours in a manner analogous to the w_{ij} values in PAMs. This time, however, the random walk probabilities are weighted by the u_{ij} of the neighbours, instead of their w_{ij}. Each of the peers that receives these messages is able to update its model of the credit topology with a path of safe credit margins starting on the message originator and ending on itself. If the peer can independently estimate its safe credit margin to any of the peers present in the credit network thus constructed, it will be able to generate a correctly oriented cycle on the credit graph, and the minimum credit on its constituent links will determine its value. Then, using a procedure analogous to the one used for contribution transfer, the peers can perform the necessary additions to their accounts in order to seed $\mathcal{G}$.

VI. Incentives Mechanism

A. Account Maintenance Incentives

Since most of the protocol elements that we propose involve the consistent modification of the account values w_{ij}, protocol stability in the presence of strategic peers demands that peers have no incentive to arbitrarily change the accounts relating to the contributions of other peers.

As detailed in Section IV-A, every time that a peer n_i sends a PAM (either newly created or forwarded) to another peer n_j, n_j can check if the value that n_i reports of w_{ji} corresponds to the actual contributions c_{ji} from n_j to n_i. Thus, the most straightforward way of dealing with this scenario would be for

n_j to reduce w_{ij} by $c_{ji} - w_{ji}$ if $c_{ji} > w_{ji}$, and to increase w_{ij} by $w_{ji} - c_{ji}$ if $w_{ji} > c_{ji}$ (as long as this does not bring $w_{ij} - w_{ji}$ over the safe credit margin u_{ij}). This policy directly mimics the reciprocity properties of the *Tit-for-Tat* trigger strategy in an Iterated Prisoner's Dilemma, and as such it can take advantage of the significant literature available [17] on its convergence, evolutionary stability, and how to deal with its vulnerability to observation noise (misjudged defections triggering mutual defection runs).

B. Protocol Incentives

The protocols described in Section IV presuppose the correct operation of the contribution accounting system, which is itself vulnerable to freeloading. To see why, it suffices to consider the benefit that a strategic peer might obtain from message forwarding. In the case of a PAM, a peer n_i does not need to forward the PAM to obtain a flow contribution path rooted on itself. Thus, by receiving PAMs but refusing to forward them, a peer can conserve its resources and still benefit from contribution topology discovery.

To provide a negative incentive to the termination of PAM walks, every peer decreases the account of the next peer in the contribution chain by an amount η before sending the PAM. Consider a PAM traversing the path in Fig. 1a. In this case, w_{ji} would be decremented by η, giving n_i a profit of η and n_j a deficit for the same amount. However, when n_j forwards the PAM to n_k and w_{kj} is decremented, n_j will regain its original level of contribution and it will be n_k who has a deficit. In this way, the deficit of η is propagated through the contribution chain, until the PAM is returned to n_i (and its initial profit cancelled). Thus, if the PAM fails to be forwarded (or returned to the originator), the last peer that received it will automatically suffer a contribution fine of η.

For CTRMs, any peers not at the contribution chain endpoints have nothing to gain from the transaction, but still have to consume their own resources. Again, any peer could choose not to propagate CTRMs from different peers, and it would save resources without impacting its own transfer capacity.

To create an incentive for contribution transfer transactions to be correctly executed, peers rely on the transaction timer t_{T}. The expiration of t_{T} triggers two events. First, the soft-state reservation for the transaction is freed. Second, each peer automatically decrements the account of the next peer in the contribution chain by an amount ζ. The account movements balance in the same way as in the PAM case, but instead of equalising the contribution contributions, the peer that failed to complete the transaction is fined with ζ, and the transaction originator is given a compensation of ζ.

VII. Attack Resistance Properties

A. Resistance to Sybil Attacks

First, we show that our proposed system is resistant to *sybil attacks* [9]. In our case, the sybil attack takes the following form: a peer n_m creates a large set of identities, and directly modifies their account values to create an arbitrary contribution network among them (a *sybil strategy*, as defined in [12]). Then, it tries to use this fictitious contribution network to extract large amounts of resources from the network.

We argue that this attack will not be profitable for the attacker. The reason for this is the equivalence between contribution transfer and maximum flow over simple paths in $\mathcal{G}$. Thus, the capacity of this "network of sybils" to extract resources from the peer-to-peer infrastructure remains bounded by the contribution values of the links that connect it to the rest of the system (the set of *attack links*, as defined in [18]). As these will be, in turn, bounded by the contributions that the peers in the sybil network make to normal peers, there is no incentive to create large numbers of identities: this will just split the contribution values amongst all of them, and no single one of them will benefit. Every one of the sybils will experience worse service than a single identity with access to the same network resources.

To analyse the sybilproofness properties of contribution transfer, we use the theoretical framework presented in [12]. We define $\mathcal{W}(s,i)$, the maximum contribution transfer capacity between a peer n_s and another peer n_i, as

$$\mathcal{W}(s,i) = \bigoplus_{P_{si} \in \mathcal{P}_{si}} \left(\bigotimes_{l_{jk}=(n_j,n_k) \in P_{si}} w_{jk} \right)$$

where $\mathcal{P}_{si} = \{P_{si}\}$ is the set of all paths from n_s to n_i, and each of the l_{jk} is a contribution link in $\mathcal{G}$. Since in our particular routing algebra $\otimes = \min$ and $\oplus = \max$ (see Section IV-C), we define $\mathcal{W}(P_{ij})$, the transfer capacity of P_{ij}, simply by applying $\min$ to the w_{nm} values of the $l_{nm} \in P_{ij}$.

From our definitions, it is easy to see that our system satisfies the conditions for both value and rank sybilproofness (Theorems 4 and 5 of [12]):

1) *Diminishing returns*. If we have a path P_{ij} with capacity $\mathcal{W}(P_{ij})$ and we concatenate it with another path P_{jk} with transfer capacity $\mathcal{W}(P_{jk})$ to create a new path P_{ik}, we have that $\mathcal{W}(P_{ik}) = \mathcal{W}(P_{ij} \cup P_{jk}) = \mathcal{W}(P_{ij}) \otimes \mathcal{W}(P_{jk}) = \min(\mathcal{W}(P_{ij}), \mathcal{W}(P_{jk})) \leq \mathcal{W}(P_{ij})$. Thus, $\otimes = \min$ is non-increasing.
2) *Monotonicity*. If we have two paths P^1_{ij} and P^2_{ij} with capacities $\mathcal{W}(P^1_{ij})$ and $\mathcal{W}(P^2_{ij})$, and we aggregate them with $\oplus = \max$, we have that $\mathcal{W}(P^1_{ij}) \oplus \mathcal{W}(P^2_{ij}) = \max(\mathcal{W}(P^1_{ij}), \mathcal{W}(P^2_{ij})) \geq \mathcal{W}(P^1_{ij})$. Thus, $\oplus = \max$ is non-decreasing.
3) *No splitting*. We consider a path $p_{in} = \{l_{ij}, l_{jk}, \ldots\}$ that has a contribution transfer capacity of $\mathcal{W}(P_{in}) = \bigotimes_{l=(n_i,n_j) \in P_{in}} w_{ij} = \min(w_{ij}, w_{jk}, \ldots)$. We split this path into two parallel paths P^1_{in} and P^2_{in} that span the same links as P_{in}, but which have link capacities w^1_{ij} and w^2_{ij} such that, for every link l_{ij}, $w^1_{ij} + w^2_{ij} = w_{ij}$. Since we require $w^1_{ij} \geq 0$ and $w^2_{ij} \geq 0$, we have that $w_{ij} \geq w^1_{ij}$ and $w_{ij} \geq w^2_{ij}$ for all l_{ij}. Thus, $\mathcal{W}(P^1_{in}) \leq \mathcal{W}(P_{in})$ and $\mathcal{W}(P^2_{in}) \leq \mathcal{W}(P_{in})$, and it follows that $\mathcal{W}(P^1_{in}) \oplus \mathcal{W}(P^2_{in}) = \max(\mathcal{W}(P^1_{in}), \mathcal{W}(P^2_{in})) \leq \mathcal{W}(P_{in})$.

Furthermore, since contribution is transferred through single routes and $\oplus = \max$, contribution transfer is rank sybilproof.

B. Resistance to Peer Slander

Our system is also designed to mitigate peer *slander*. One case of this problem involves peers lying about the contributions that they have received from other peers. As noted in Sections IV-A and IV-B, the effectiveness of this attack is reduced because all information concerning the contribution of a peer must be vetted by its neighbours before being forwarded. Another, related case involves peers lying about the contributions that they have given to other peers. Again, this is of limited impact in our system, since it would entail forging a digital signature.

C. Resistance to Whitewashing

Finally, we show that our system is resistant to *whitewashing* attacks: the creation of disposable identities, so that any penalties that the system might have imposed on misbehaving peers are "forgotten". Since the only way to make whitewashing unprofitable is to make a newcomer and a heavily punished node indistinguishable, this attack is difficult to defend against: it exploits altruistic system characteristics that are desirable for bootstrapping and incorporating new peers into the system.

In our system, all punishments are equivalent to the loss of previous contributions. Thus, both newcomers and heavily punished nodes have zero social capital. Since these nodes are unable to engage in reciprocative interactions, they will only receive best effort service (they will be easily preempted by peers with higher contribution values) and will not be able to participate in the contribution transfer network. To see why, it suffices to recall that the PAM random walk is weighted by previous contributions: a node with very small previous contributions has a very small probability of being forwarded a PAM. As nodes increase their contributions, they become more heavily embedded in the contribution network and their capacity for contribution transfer increases. However, we note that while non-malicious peers will only need to build up their contributions once, whitewashers will need to do so several times: once for each one of their new identities. Thus, peers have an incentive to maintain their identities and behave correctly towards other peers.

VIII. Evaluation

We analyse our system by performing extensive simulations. When we use a synthetic contribution topology $\mathcal{G}$ to test our protocols, we use Erdős-Rényi [19] graphs, and if weighting is required, we assign link weights following a uniform distribution. We simulate each peer individually, in order to take into account differences between the true contribution network $\mathcal{G}$ and its sampled model on each peer, $\mathcal{G}_i$.

A. Contribution Topology Discovery

In order to get a better understanding of the behaviour of our accounting mechanism, we track the formation of the contribution network subgraph $\mathcal{G}_i$ for each of the peers as the PAMs propagate through the contribution network. To evaluate this, we use a static $\mathcal{G}$ with 1000 peers, each with an average of 150 neighbours. We leave the analysis of the influence of dynamic changes to these parameters for further work.

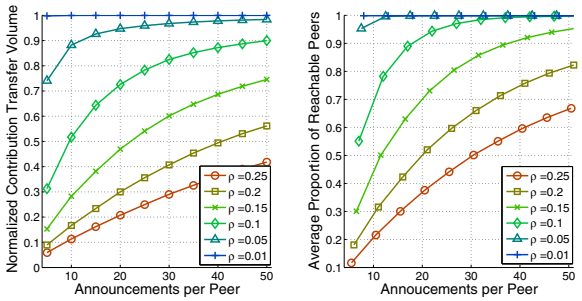

Fig. 2. Contribution transfer volume and reachability in $\mathcal{G}_i$

As expected from our proposed PAM forwarding probability (1), the locally constructed subgraphs $\mathcal{G}_i$ tend to be much sparser than $\mathcal{G}$ and thus may fail to account for all possible paths through which contribution transfers could be routed. To analyse this, we consider the *normalised contribution transfer capacity*: the sum of all the potential contribution transfers that n_i could perform over $\mathcal{G}_i$ using the routing algebra of Section IV-C, normalised by the same magnitude calculated over the entire contribution network $\mathcal{G}$. As we can see in Fig. 2 (left), not many messages are needed to attain a high capability for contribution transfer. With just 10 PAMs per peer and a ρ_{stop} of .1, the average contribution transfer capacity of a peer is nearly at 55% of its theoretical maximum.

If we focus not on the capacity of the contribution transfers, but on the existence of a contribution transitivity path on $\mathcal{G}_i$, our results are even better. As shown in Fig. 2 (right), the proportion of peers reachable through contribution transfers grows quickly with the number of PAMs per peer (its growth rate decreases, as expected, as the walk length approaches the graph diameter). The standard deviation of the distribution of both the normalised contribution transfer capacity and the number of peers reachable through the discovered contribution transfer paths decreases with increasing PAMs per peer, as $\mathcal{G}_i$ approximates $\mathcal{G}$.

It is interesting to consider the cumulative distribution function (CDF) of the capacities of all the contribution transfers that peers can achieve using their locally constructed $\mathcal{G}_i$. We compare it with the CDF of all the contribution transfer capacities that could be achieved if every peer had perfect knowledge of $\mathcal{G}$, represented as a dashed line in Fig. 3 (due to construction of our simulated $\mathcal{G}$, most peers tend to achieve roughly the same contribution transfer capacity between them). As the number of PAMs increases, $\mathcal{G}_i$ samples $\mathcal{G}$ more accurately and the CDF of locally discovered contribution transfer capacities becomes much closer to the theoretical maximum.

B. Contribution Transfer and Indirect Reciprocity

To evaluate the properties of our system as an incentives mechanism,we use an *interest digraph* $\mathcal{H}$ to model peer service preferences: an edge (n_i, n_j) in $\mathcal{H}$ implies that peer n_i is interested in requesting services from peer n_j. We define

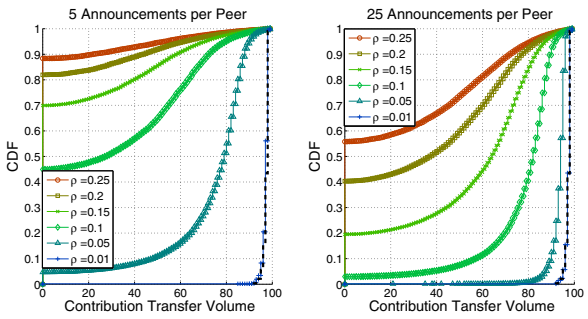

Fig. 3. Change in transfer capacity CDF with number of PAMs

two types of peers: *reciprocators*, that follow the reciprocity rules described in Sections IV, and *freeloaders* that rely exclusively on altruism and refuse to honour any service requests. Moreover, we distinguish two kinds of reciprocators: *enabled* nodes are able to transfer contributions and perform contribution seeding as described in Sections IV and V[3], while *non-enabled* nodes can only perform direct reciprocity (strict *Tit-for-Tat* with nodes they have interacted with in the past).

In order to focus exclusively on the reciprocity properties of the system, we neglect peer capacity: all peers will respond to all requests either with high quality service, best effort service, or refusal due to freeloading behaviour. We are interested in exploring how the service quality experienced by reciprocators and freeloaders varies as $N_{\mathcal{H}}$, the average peer outdegree in $\mathcal{H}$, and $\frac{R}{F}$, the proportion of reciprocators vs. freeloaders, change.

We start each *simulation run* with nodes having no contributions between them, and then allow the simulation to run until an equilibrium is reached where the average proportion of services that a peer receives through reciprocation and altruism remains unchanged. On each one of the epochs of a simulation run, the actions in Algorithm 1 are performed, where R_1, R_2 and *random* represent selection strategies for each peer n_i:

- R_1: Prefer peers to whom n_i has contributed to, and are thus indebted to it (this include contribution transfer recipients).
- R_2: Prefer peers that have reciprocated n_i in the past.
- *random*: Choose any peer randomly.

Finally, we run a simulation run for each possible combination of $N_{\mathcal{H}}$ and $\frac{R}{F}$. The results are shown in Fig. 4.

In Fig. 4, blue circles represent *reciprocators* and red triangles represent *freeloaders*. The left column shows the results obtained when *reciprocators* are *non-enabled*, while the right column shows those obtained when *reciprocators* are *enabled*. Finally, the top row shows the proportion of time that peers obtain high priority service, while the bottom row shows the proportion of time that peers obtain best effort service[4]. We plot the average service quality that a peer receives, as a

[3]When an *enabled* peer requests a service from a peer to whom it has not contributed, it will attempt to execute a contribution transfer operation. When contacted by a peer to whom they owe past contributions, *reciprocators* will provide prioritised service (all other requests will only get best-effort service). When a freeloader receives a request, it will ignore it.

[4]Due to freeloaders, some requests will not be honoured and these two rows will fail to add up to one.

Algorithm 1 Execute *simulation epoch* for single time t

for all $n_i \in \mathcal{G}$ (in random order) **do**
  for all service requests s_i of n_i (10 in our simulation) **do**
    Calculate S_i, the set of peers known to n_i of interest in $\mathcal{H}$ ($S_i \subset \mathcal{G}_i$, $S_i \in \Pi^-(n_i)$ in $\mathcal{H}$)
    Rank all $n_s \in S_i$ using R_1, then tie-break ranking with R_2, then tie-break with *random*
    Send request s_i to the peer n_s that had the best ranking
  end for
end for

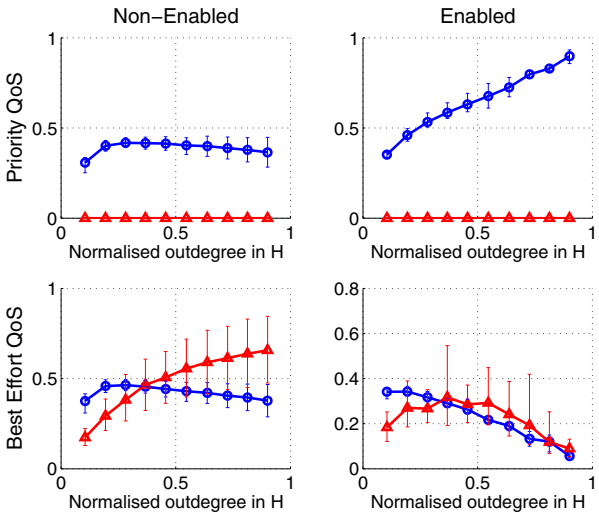

Fig. 4. Average Service Quality as a function of $\frac{N_{\mathcal{H}}}{|\mathcal{H}|}$

function of $N_{\mathcal{H}}$ normalised by the number of peers in $\mathcal{H}$. The error bars mark the maximum and minimum values obtained by varying $\frac{R}{F}$ between .5 and .9, for each value of $N_{\mathcal{H}}$.

We can see that both *enabled* and *non-enabled* reciprocators obtain a higher ratio of prioritised service as $N_{\mathcal{H}}$ increases, since this gives them a better chance of being able to find a peer to whom they can send a reciprocating request. However, *enabled* peers experience a much higher proportion of prioritised service requests, and this proportion increases much more quickly than for *non-enabled* peers. This is because contribution transfer allows *enabled* nodes to perform reciprocative interactions almost exclusively, being able to route their contributions in $\mathcal{G}$ to match the supply and demand constraints imposed by $\mathcal{H}$. Unable to do this, *non-enabled* peers are forced to maintain higher proportions of their service requests as altruistic ones, thus suffering a service quality penalty. Additionally, we see that freeloaders experience consistently worse service levels and much higher variability, thus validating the effectiveness of reciprocity as an incentives mechanism.

IX. Related Work

A fundamental characteristic of indirect reciprocity is that it requires a much greater degree of social structure and organ-

isation. On [20] the authors analyse the effect of BitTorrent *communities* on the cooperative behaviour of users.

In essence, community sites give peers the benefits usually associated with social capital: information diffusion, access to privileged knowledge, and increased trust in transaction outcome. Sites impose their own set of "social rules", and noncompliance is punished with denial to access to social capital (community resources). The authors show that this risk of losing social capital is enough to change the incentive structure of BitTorrent users, which then change their behaviour to a more reciprocally altruistic one.

Usually, sharing in peer-to-peer networks is done in terms of mutually useful content, and thus the problem of content search, distribution and replication becomes part of the incentives problem. In [21], the formation of *alliance groups* is proposed as a way to alleviate this situation. Basically, the authors modify the BitTorrent protocol so that peers aggregate in trusted groups where peers download fragments on behalf of each other. They name their approach *amortised Tit For Tat*, because the allies collect fragments on behalf of the collector only out of a reciprocal altruism expectation. In this case, peers use their resources to download content for which they had no interest, in order to "store" this contribution and use it later to download other content. This is, thus, an implementation of time-deferred indirect reciprocity, but only within the peers in the alliance group. **PledgeRoute** allows reciprocity with arbitrary peers at arbitrary times without the need of trusted groups.

Finally, in [22], the authors propose a indirect reciprocity scheme based on n-way cyclical exchanges. They propose an algorithm where a request tree is formed, and for requests that connect disjoint branches of the tree, a cycle is identified through the common ancestor of both branches. The authors show that the scheme provides superior performance to direct reciprocity. However, their technique does not support time-deferred reciprocity.

X. Conclusions

In this paper we presented **PledgeRoute**, a system that enables overlay network peers to contribute resources to a set of peers and use these contributions to obtain resources from a different set of peers at a different time. The system is resistant to sybil, slandering and whitewashing attacks, and operates as an incentives mechanism by ensuring higher service quality for those peers that contribute more resources to the overlay.

We achieved this by analysing the contribution transfer operation using a routing algebra with sybilproofness properties similar to those of maximum flow. To make the system scalable, we proposed a topology sampling algorithm that extracts sparse subgraphs with high contribution transfer potential.

Since the contribution transfer operation tends to reduce the absolute amount of contributions in the network, we proposed a contribution seeding algorithm based on the detection of credit cycles, which can the be used to infuse the network with social capital through virtual contributions between peers.

Finally, we presented incentive schemes that work alongside the aforementioned techniques in order to ensure the correct operation of the protocol in the presence of strategic peers. When coupled with simple reciprocity policies, **PledgeRoute** allows contributing peers to obtain higher service quality than freeloaders, thus becoming an incentives mechanism suitable for peer-to-peer networks.

Acknowledgements

The authors would like to thank John Billings and Alexander Gurney for their discussions on routing algebras, and the anonymous reviewers for their helpful reviews. This work was partly supported by CONACYT award 130561.

References

[1] K. Lai, M. Feldman, I. Stoica, and J. Chuang, "Incentives for cooperation in peer-to-peer networks," in *Proceedings of P2PEcon*, 2003.

[2] E. Adar and B. Huberman, "Free riding on gnutella," Xerox PARC, Tech. Rep., August 2000.

[3] M. Piatek, T. Isdal, T. Anderson, A. Krishnamurthy, and A. Venkataramani, "Do incentives build robustness in bittorrent?" in *Proceedings of NSDI'07*, Cambridge, MA, April 2007.

[4] R. Axelrod and W. D. Hamilton, "The evolution of cooperation," *Science*, vol. 211, March 1981.

[5] B. Cohen, "Incentives build robustness in bittorrent," in *Proceedings of P2PEcon*, Berkeley, CA, USA, 2003.

[6] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The EigenTrust algorithm for reputation management in P2P networks," in *Proceedings of WWW '03*. New York, USA: ACM Press, 2003, pp. 640–651.

[7] C. Dellarocas, *Reputation Mechanisms*, T. Hendershott, Ed. Elsevier, 2006.

[8] R. Gupta and A. K. Somani, "Reputation management framework and its use as currency in large-scale peer-to-peer networks," in *Proceedings of P2P'04*. Washington, DC, USA: IEEE Computer Society, 2004, pp. 124–132.

[9] J. R. Douceur, "The sybil attack." in *Proccedings of IPTPS 2002*, Cambridge, MA, March 2002, pp. 251–260.

[10] Z. Zhang, S. Chen, and M. Yoon, "MARCH: A distributed incentive scheme for peer-to-peer networks," in *Proceedings of INFOCOM 2007*, May 2007, pp. 1091–1099.

[11] F. D. Garcia and J.-H. Hoepman, "Off-line karma: A decentralized currency for peer-to-peer and grid applications," in *Proceedings of ACNS 2005*, vol. 3531, 2005, pp. 364–377.

[12] A. Cheng and E. Friedman, "Sybilproof reputation mechanisms," in *Proceedings of P2PEcon '05*. New York, USA: ACM, 2005, pp. 128–132.

[13] J. S. Coleman, "Social capital in the creation of human capital," *The American Journal of Sociology*, vol. 94, pp. S95–S120, 1988.

[14] R. L. Trivers, "The evolution of reciprocal altruism," *Quarterly Review of Biology*, vol. 46, p. 35, 1971.

[15] T. G. Griffin and J. L. Sobrinho, "Metarouting," in *Proceedings of SIGCOMM '05*. New York, NY, USA: ACM, 2005, pp. 1–12.

[16] R. L. Rivest and C. E. Leiserson, *Introduction to Algorithms*. New York, NY, USA: McGraw-Hill, Inc., 1990.

[17] G. Kendall, X. Yao, and S. Y. Chong, *The Iterated Prisoner's Dilemma: 20 years on*. WSPC, 2007.

[18] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman, "Sybilguard: defending against sybil attacks via social networks," in *Proceedings of SIGCOMM '06*. New York, NY, USA: ACM Press, 2006, pp. 267–278.

[19] B. Bollobas, *Random Graphs*, W. Fulton, A. Katok, F. Kirwan, P. Sarnak, B. Simon, and B. Totaro, Eds. Cambridge University Press, 2001.

[20] N. Andrade, M. Mowbray, A. Lima, G. Wagner, and M. Ripeanu, "Influences on cooperation in BitTorrent communities," in *Proceedings of P2PEcon '05*. New York, NY, USA: ACM Press, 2005, pp. 111–115.

[21] P. Garbacki, D. H. J. Epema, and M. van Steen, "An amortized tit-for-tat protocol for exchanging bandwidth instead of content in p2p networks," in *Proceedings of SASO 2007*, Boston, MA, July 2007.

[22] K. G. Anagnostakis and M. B. Greenwald, "Exchange-based incentive mechanisms for peer-to-peer file sharing," in *Proceedings of ICDCS'04*. Washington, DC, USA: IEEE Computer Society, 2004, pp. 524–533.