Altruism, reciprocity, and tokens to reward forwarding data: Is that fair?

Vahid Heidaripour Lakhani[1], Arman Babaei[2], Leander Jehl[1], Georgy Ishmaev[3], Vero Estrada-Galiñanes[2]
[1]*University of Stavanger*, Stavanger, Norway
[2]*École Polytechnique Fédérale de Lausanne (EPFL)*, Lausanne, Switzerland
[3]*Delft University of Technology*, Delft, Netherlands

***Abstract*—Decentralized storage networks offer services with intriguing possibilities to reduce inequalities in an extremely centralized market. Fair distribution of rewards, however, is still a persistent problem in the current generation of decentralized applications using token-based incentives. They are often disproportionally concentrated with small number of early adopters and high-resourced participants. Incentive mechanisms capable of addressing this problem are still poorly understood. This paper aims to help fill this gap by developing our Tit-for-Token (Tit4Tok) model. Tit4Tok realizes incentives based on the triad of altruism (selfless behavior), reciprocity (Tit-for-Tat), and monetary rewards compatible with a free market. Tit4Tok analyzes the effects of storage-, and network-parameters fine-tuning to achieve fair distribution of rewards for participants.**

We present a comprehensive exploration of different factors when incentivized peers share bandwidth in a libp2p-based network, including uneven distributions emerging when gateways provide data to users outside the network. We quantified the Income-Fairness with the Gini coefficient, using multiple model instantiations and diverse approaches for debt cancellation. We propose regular changes to the gateway neighborhood and show that our shuffling method improves the Income-Fairness from 0.66 to 0.16. We quantified the non-negligible cost of tolerating free-riding (altruism). The performance is evaluated by extensive computer simulations and using an IPFS workload to study the effects of caching.

***Index Terms*—Fairness, Bandwidth Incentives, Token-based Incentives, Networked Economy, Web3 Incentives, Reciprocity, Tit-for-Tat, Monetary-based Incentives, Decentralized storage networks, Prefix-based Routing Networks**

I. Introduction

Open decentralized systems (ODS) such as the Interplanetary File System (IPFS) or the Swarm network propose a tantalizing vision of a decentralized web and a fair data economy through large-scale collaborative ecosystems. These data-sharing platforms facilitated by peers moving and storing data across a network depend on effective incentives for users, content generators, and operators [1], [3], [12], [25]. There is, however, a deficit in understanding how incentive mechanisms can provide fair distribution of rewards to participants, i.e., minimizing the differences between profits from equally entitled agents. This a critical issue since lack of fairness undermines self-sufficiency and leads to the concentration of rewards with few highly-resourced participants, as we know from the experience of Web3 decentralized apps that use token-based rewards [17].

This research is partially funded by the BBChain and Credence projects under grants 274451 and 288126 from the Research Council of Norway.

Thus, better understanding of fair incentives is a necessary enabling factor for ODS. This is desirable as evidence shows that despite some problems with the reliability and manageability of autonomous peer operators forming peer-to-peer (p2p) networks, these networks provide very cost-effective solutions. This is evident from the observation that even large companies benefit from these networks, especially in edge-computing, content distribution networks, and other systems using mechanisms similar to those used in the BitTorrent network [8], [14], [19], [23], [51].

Many ODS, including the systems mentioned above, have a networking stack that depends on the libp2p project [2]. This modular library provides key components to build decentralized networks equally accessible from anywhere in the world. On top of the networking layer, many systems include some kind of incentivization layer. For instance, Swarm uses the SWAP protocol and the postage stamps to incentivize bandwidth and storage sharing respectively. The topic of incentives has received a lot of attention in systems research in part thanks to the advancements in blockchain protocols and token-based incentivization mechanisms [6], [15], [24], [41], [55]. Nonetheless, the literature is vague when it comes to discerning how the resources shared in the network are financed and if peer operators receive a fair reward for contributing to the ecosystem. We know that imbalances in incentives can cause centralization problems, e.g., consensus power concentration, routing centralization, wealth concentration, bandwidth concentration, etc. However, a taxonomy of centralization in public systems [42] showed that consensus power has been widely studied, leaving, in comparison, a research gap for the factors that affect bandwidth incentives. This paper aims to help fill this gap by developing *Tit-for-Token* or, in short, *Tit4Tok*, a framework to understand fairness when forwarding data by incentivized peers in decentralized storage networks. We use *Tit4Tok* to investigate potential sources for uneven token distributions and their effect on fairness. For example, one of the sources of uneven distributions is when a few peers are originating large amounts of requests. This situation can happen when peers act as a gateway to take requests from clients that do not participate in the network and access a gateway, for example, via a normal browser.

Incentives often play a significant role in motivating peer operators to participate in a network. Well-designed incentives can be used to reinforce motivation. We define the *triad of altruism, reciprocity, and free enterprise* as required incentives for a more fair data economy. In other words, our framework realizes a triad that comprises acts of kindness such as debt forgiveness, mirroring cooperation such as in standard Tit-for-Tat incentives, and monetary rewards as desired in the free market. We think that realizing this triad could potentially bring closer the vision of a fair data economy found in the Swarm network community. While the ideas of a fair data economy, data sovereignty, and decolonization of the digital space are flourishing among society, e.g., software developers, content creators, artists, investors, and EU policymakers, the literature lags behind. This is unfortunate for advancing networked systems research with societal impact. The majority of the papers on incentives published during the last three decades only focus on one or at most two aspects of the triad [24]. For an illustrative example, we need to retrace incentives to the reciprocity of the Tit-for-Tat (from here on abbreviated *Tit4Tat*) found in BitTorrent [11]. Despite the success of BitTorrent, and the good arguments about *Tit4Tat*, this mechanism, or many others that have been proposed later, is not solid enough to provide an alternative to the asymmetric wealth distribution of the current data economy. While our *Tit4Tok* model is inspired by the Swarm network, we are the first to formalize and analyze this model. Our evaluation shows, for example, that with a naive parametrization, altruism and reciprocity may significantly hurt fairness in reward distribution. Our main contributions are:

Tit4Tok model: We introduce an abstract model to incentivize network peers to realize the triad of altruism, reciprocity, and free enterprise.

Fairness analysis: We measure the Gini coefficient to quantify the income fairness on debt balances and token transfers. We conduct extensive simulations to study mechanisms for altruism and reciprocity under different network settings and instantiations. We evaluate both novel mechanisms, and settings used in real systems like Swarm, which have not been analyzed previously. Focusing on the case where gateways provide data to users outside the network we that a novel instantiation, as well as connection shuffling can mitigate the reduced fairness, caused by altruism and reciprocity.

II. Tit-for-Tat: Preliminaries and Prior Work

P2p networks are computationally-based human social systems, establishing a shared resource pool within an open and often permissionless community. However, an unregulated commons poses the risk of resource overuse, potentially leading to a collapse known as the "tragedy of the common" [22]. Thus, managing computational resources in these open networks is vital, necessitating effective incentive mechanisms.

A. Tit-for-Tat: Strength and Limitations

The *Tit4Tat* mechanism has been intensively used in p2p networks, e.g., BitTorrent [11], with the general belief that it efficiently discourages free-riding, i.e., peers who only consume resources without giving back. It has a simple strategy in which each participant first cooperates and then mirrors, or reciprocates, the immediately observed behavior from its interacting peers, likely to incentivize mutual cooperation. *Tit4Tat* works by punishing bad behaviors in the future, i.e., *cheat me first, and I will cheat you back*. This concept, referred to as the "shadow of the future," was largely studied by Axelrod in computer tournaments playing the Prisoner's Dilemma cooperation game [5]. Later, the Pavlov strategy proposed a more robust strategy that included some degree of forgiveness or generosity between peers [36].

Altruism vs Free-Riding. The cooperation that surges from *Tit4Tat* is known as "reciprocally altruistic behavior" [47]. But does it work in practice? One of the main criticisms is that *Tit4Tat* can be easily subverted by participants who change their client's code to cheat (fail to reciprocate) their peers. The *Tit4Tat* in BitTorrent can induce free riding [27], and entire files can be downloaded without reciprocating in a cheap free-riding attack [30], [38]. The problem of selfish and misbehaving nodes was widely studied in the literature, which offers a plethora of strategies, including punishments and/or variations of the *Tit4Tat* to mitigate misbehaviors [16], [18], [27], [29], [31], [37], [53], [54]. On the contrary, altruistic behavior provides an alternative narrative, which explains why networks do not collapse [50]. Moreover, a generalized reciprocity behavior, in which peers do favors without direct expectations while relying only on somebody else willing to do a favor to them, can explain why peers can tolerate some free-riding behavior [26]. Tribler presented down-to-earth expectations about altruism with its social group incentives based on the "kinship fosters cooperation" argument [39]. Scientists found that even a small amount of altruism effectively improved the performance of a p2p live streaming service [10]. As other scientists noted, substantial research has been dedicated to discouraging selfish behavior with complex technical solutions or disregarding the cost overhead [13]. Considering all the above, our model *Tit4Tok*, which builds on *Tit4Tat* and real-world decentralized networks, motivates altruistic behaviors continuously and tolerates some free-riding.

Mechanism Dependencies on Upper Layers. *Tit4Tat* reciprocates the immediately observed behavior, and going beyond that requires a public history of behaviors, a robust identification layer, or other complex mechanisms often found in reputation-based incentives. The overarching question is which peer is trustworthy or at least offers the best cost-quality service relationship [45]. Reputation-based systems present multiple challenges and can harm participants. For example, "Sybil attacks," in which a single entity controls multiple fake identities either to inflate its reputation value or discredit other participants [40]. Despite the centralized control, even YouTube cannot stop abusers from illicit monetization exploits like selling accounts [9]. Our paper focuses on the networking and incentive layers without adding dependencies to more complex mechanisms. *Tit4Tok* has a mutual accounting layer that could be further improved with ideas such as indirect reciprocity [35] to address manipulation by Sybils without the cost of a global reputation layer.

Novel Business Models. *Tit4Tat* drives peers in BitTorrent to exchange content of mutual interest. Even if reciprocating bandwidth instead of content is an improvement [18], the model behind it is a restricted version of a barter economy that suffers from the double coincidence of wants, i.e., peers would reward a forwarded chunk with another forwarded chunk. Thus, *Tit4Tat* impairs complex business developments. Monetary or credit-based incentives can address this by enabling more practical transactions among peers than in bartering economies. This area has generated significant interest [20], [28], [33], [44], especially with recent research focusing on cryptocurrencies and token-based incentives. *Tit4Tok* allows to transfer tokens to settle debt and thus also encourages peers to participate for monetary profit. However, *Tit4Tok* does include altruism and reciprocity, avoiding the overhead of settling every transaction.

B. Literature Gap

A recent survey on p2p incentive mechanisms, reviewed publications between 1993 to 2022, reported a large number of papers on *Tit4Tat* and auction-based monetary incentives, but negligible focus on the interplay between DHTs and incentives [24]. Further, reproducing literature results is challenging due to unavailable or discontinued simulation tools. Regrettably, societal and free service aspects remain neglected, e.g., the term “free,” encountered over 60 times, is solely associated with free-riding mitigation schemes.

We argue that previous studies on incentives in decentralized networks have not adequately explored the integration of altruism, reciprocity, and free enterprise. Our paper differs significantly from prior studies in this area by introducing an incentive model that allows for limited free-riding. This model aligns with societal concerns and common strategies for increased adoption, as it supports freedom of expression, promotes universal access, and fosters desirable network effects. Moreover, our model is more cost-effective than previous models that rely on anti-free-riding measures.

III. Decentralized Storage Networks

Terminology. *Nodes* (or equivalently *peers*) in a p2p network connect with *neighbors*, sharing bandwidth to efficiently transfer data between *users* (consumers) and *storers* (suppliers). Users can operate their own peers (*originators*) within the network or opt for third-party *gateways*, acting as interfaces between the decentralized network and external entities. Gateways receive external requests and initiate requests on behalf of users. Local content (*chunks*) stored in a node becomes accessible to others once pushed into the network. Storer nodes are accountable for chunks falling within their responsibility area. Content transfer occurs via *routing paths* and *forwarding actions* among peers until reaching the destination, benefiting users, creators, and stakeholders while enabling paid storage services for storers. Bandwidth, therefore, emerges as a crucial resource in decentralized storage networks.

A. Peer-to-Peer Networking Stack

The libp2p library [2] by Protocol Labs provides a versatile framework utilized in various decentralized storage systems such as the InterPlanetary File System and Swarm. Its Kad-DHT component, based on Kademlia [32] and S/Kademlia [7], enables efficient peer and content discovery.

The Kademlia Distributed Hash Table (DHT) minimizes hops to reach network peers while requiring only a small number of direct connections per peer. Routing tables are organized based on a *distance metric*, employing bitwise XOR of hashed keys. The routing table of a peer v contains buckets, each containing up to k peers at the same distance to v. Peers close in the address space form a *neighborhood* and are responsible for storing multiple chunk replicas.

B. Swarm and its SWAP Protocol

Swarm is a decentralized network that enable peers to provide global storage and communication services; its mission is to foster a fair data economy [3].

Networking Layer. Initially integrated with Ethereum’s devp2p network, Swarm later adopted the libp2p library, facilitating content distribution through incentivized peers. The network employs forwarding Kademlia for chunk sync and retrieval [48]. In forwarding Kademlia, an originator initiates a request that is relayed via forwarding nodes $F_0, \ldots, F_n$ all the way to storer node S, the storer node closest to the chunk address. The chunk is then delivered by being passed back along the same route to the downloader. It ensures content-addressing granularity and provides some degree of ambiguity and deniability for peers sending chunks.

Incentives Layer. Swarm’s incentives layer encompasses bandwidth incentives for fast data provision and storage incentives for long-term preservation, using BZZ tokens bridged via ERC-20. Development iterations, such as Swap, Swear, and Swindle protocol [49], combine off-chain communication and settlements with on-chain enforcement through smart contracts on the Gnosis Chain blockchain. The bandwidth incentive uses off-chain payments by accumulating the debt and issuing a cheque that can be cashed out at a later time. The cashing out process and the storage incentive take place on-chain.

Business Layer. Swarm Foundation’s goal is to create a self-sustaining infrastructure for a data supply-chain economy, enabling developers to create decentralized applications (dApps) and media files without hosting costs, fostering decentralization and inclusivity [48].

IV. Tit-for-Token

A. Design Overview

Our design is based on pairwise credit accounting, with a possibility for debt forgiveness and token payments that work together in a layered architecture. Figure 1 presents an overview of *Tit4Tok* with its network, accounting, and settlement layers (the exchange layer is excluded from *Tit4Tok* but provided for completeness).

On the *network layer* peers $\{p_n; n \in N\}$ form a self-organized overlay using a DHT table and route requests

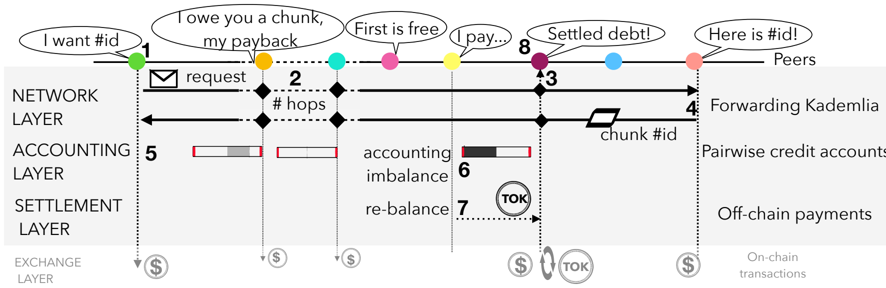


Figure 1: *Tit4Tok* includes routing and accounting layers. The settlement layer transfers tokens through off-chain payments. The exchange layer is excluded. Circles positioned along the "Peers" line, each with a text bulb indicating communication, represent peers on a route from downloader to storage. 1) A gateway requests a chunk, and 2-4) forwarding operators route the request to a storer and the chunk back to the gateway, sometimes for free or in reciprocity for services. 5) Pairwise balances are updated, and 6) an unbalanced account is 7) rebalanced through token transfer. 8) Tokens are sold on the exchange layer to extract monetary rewards.

using forwarding Kademlia. Peer connections are bidirectional, meaning if peer p_1 is part of p_2's routing table, then also p_2 is part of p_1's routing table.

On the *accounting layer*, peers maintain pairwise balances associated with each connection in the routing layer with a threshold. Pairwise balances are not verifiable to others and, thus, do not need transactional or enforcement mechanisms. When receiving a chunk from a neighbor, balances are updated, and the sending neighbor is credited some accounting units. This happens independently on every hop in the route. We note that each peer p_i performing a forwarding action is credited some amount c_i^{in} from the previous peer on the route, and again credits some amount c_i^{out} to the next peer. We say that the difference between these two amounts $c_i^{in} - c_i^{out}$ is the *reward* p_i receives for forwarding. The reward for storing is simply the credited amount. We note that the reward for a storing action is incentivizing bandwidth usage. The incentivization of long-term storage is done through storage incentives, which are out of the scope of this paper.

On the *settlement layer*, peers transfer utility tokens to each other. Settlements are used to rebalance pairwise accounts and may result in a monetary transfer. We assume settlements are realized through an off-chain payment, similar to Swarm's signed receipts. We also assume the existence of an exchange layer, where utility tokens received through a transfer can be converted to native currency.

B. Debts: Forgiveness and Payments

The accounting layer enables reciprocity via pairwise credit accounting balances. Peers accumulate debts with their neighbors in their respective pairwise accounting balances. The services that peers provide to each other may balance out without requiring settlement via utility tokens. This way, peers can provide service without requiring settlement as long as the pairwise balance is below a given threshold. If not, there are two possibilities: 1) once the debt hits the threshold, the indebted peer can pay tokens to reduce the debt through the settlement layer, and the creditor receives a monetary reward; 2) once the debt hits the threshold, the indebted peer can wait for a refresh rate to get the debt reduced unless the creditor rejects the connection. If a peer always waits for the second option, it receives a limited service for free.

C. System Model and Assumptions

Normal Operation. We use *Tit4Tok* in our toolkit to study how the tokens paid by originators get distributed in a well-functioning network. We assume no significant churn, massive failures, or massive amounts of Sybils exist in the system.

Selfish (rational) clients. We assume gateways aim for profitability within the network, utilize maximum free service and pay for requests surpassing the threshold through token transfers. While chunk uploads operate similarly to downloads and use the same layers, our focus remains on chunk retrieval due to its higher frequency. In Swarm, chunk uploads contribute to storage incentives. We assume a storage incentive in Swarm is in place, a mechanism similar to [46] while we do not consider it in our analysis. We do not model the exchange layer or the detailed mechanisms of token transfers but assume they ensure finality and incur no fee or added cost. Such assumptions can be approximated by off-chain solutions where fees can be amortized over many individual settlements [21].

We assume there is a fixed exchange rate from utility tokens to accounting units and assume that utility tokens have a stable market price reflecting their usefulness. In the rest of this paper, we mostly ignore utility tokens. Instead, we say that a certain amount of accounting units is settled if the corresponding amount of utility tokens are being transferred.

We assume there is a fixed exchange rate from utility tokens to accounting units and assume that utility tokens have a stable market price reflecting their usefulness. In the rest of this paper, we mostly ignore utility tokens. Instead, we say that a certain amount of accounting units is settled if the corresponding amount of utility tokens are being transferred.

V. Fairness

A. Fairness in ODS

The notion of fairness is generally rather broad and multi-faceted, even when considered in a specific context of ODS.

It can refer either to participants' perceived fairness of a system (and its components) or to some observable agreed-upon metric. While both of these aspects are important for ODS design, here we focus on fairness in the latter sense.

Wierzbicki offered an interdisciplinary view of fairness in ODS[1]. Social psychologists judge fairness mainly through three perspectives: distributive, procedural, and retributive fairness. Distributive fairness is about the minimization of the differences between the shares or profits that equally entitled agents collect from a system. As an example, operators willing to contribute similar resources to the system, i.e., running nodes on a similar infrastructure, should receive the same token rewards. Procedural fairness refers to impartial and fair processes, e.g., having transparent and impartial mechanisms that reward peers with honest behavior. Retributive fairness is about the proportion between sanctions and rule violations, i.e., ensuring that rule-breakers are fairly treated.

The main focus of this paper lies in distributive fairness, and we define a suitable metric to measure it below.

B. Measuring fairness

We define Income-Fairness as a metric for distributive fairness in decentralized networks. Income-Fairness is the Gini coefficient computed over the net income different peers received during a given time frame. It assesses the evenness of token distribution among peers. While this metric of inequality has certain limitations in the context of complex macroeconomic models [52], it does provide helpful insights for the measurements of decentralized systems [43]. The Gini coefficient ranges from 0 (best) to 1 (worst). The best Income-Fairness suggests equal total income among peers, ensuring fair rewards for resources provisioned. Income-Fairness is relative, thus, experiments where a different amount of tokens is spent may have the same Income-Fairness, if tokens are distributed similarly. Income-Fairness can be evaluated at the accounting or settlement layer, counting reward allocation (accounting units) or transaction settlements (tokens) respectively.

VI. *Tit4Tok* Implementation

A. How many accounting units for a chunk?

We consider two models for the credit peers receive for replying to a request: constant reward and distance-based credit. This happens on the accounting layer of our model. Remember that for forwarding actions, reward $c_i^{in} - c_i^{out}$ is the difference between accounting units credited to a peer p_i and by p_i to the next peer. Here $c_i^{out} = c_{i+1}^{in}$.

Constant reward. In this model, all peers on the path of a request receive the same reward. This model is not practical but is a useful baseline for our evaluation. With constant reward, the income a peer receives is proportional to the number of answered requests.

Distance-based credit. This model uses the XOR distance to find the accounting units credited for a chunk. The credit c_i^{in} is calculated based on the distance of peer p_i from the chunk. The net credit received by p_i, $c_i^{in} - c_i^{out}$ is determined by the distance over which p_i forwards the request and reply. In this way, peers are motivated to forward to the peer with the shortest possible distance to the destination.

Distance-based credit is used in the Swarm network. As the Swarm network, we use Equation 1 to determine c_i^{in}. Here $\texttt{commonBits}(p_i, \mathit{chunkAddress})$ returns the number of bits in the common prefix of the two addresses. The constant $+1$ in the equation ensures that the last peer in the route (performing the storage action) receives at least one accounting unit. ω is a configurable parameter. When sending a request to peer p_i, $\texttt{commonBits}(p_i, \mathit{chunkAddress})$ is at least 1. Thus ω is the maximum amount for c_i^{in}. Therefore, ω is a useful unit also for other parameters in pairwise accounting. We note that in Swarm, c_i^{in} is additionally multiplied by a constant *price*, which we omit for simplicity.

$$c_i^{in} = (\max(0, \omega - \texttt{commonBits}(p_i, \mathit{chunkAddress})) + 1) \quad (1)$$

We could find no information about how to initialize ω in Swarm documents. We evaluate different parameters for ω but keep $\omega \geq \delta$, where δ is the storage depth, another parameter determining whether a peer is responsible for storing a chunk. A peer p_i is responsible to store a chunk if $\texttt{commonBits}(p_i, \mathit{chunkAddress}) \geq \delta$. Setting $\omega \geq \delta$ thus ensures $\omega > \texttt{commonBits}(p_j, \mathit{chunkAddress})$ for a peer p_j not storing a chunk. Peer p_j can forward the request to a different peer p_{j+1} closer to the chunk, and receive $c_j^{in} - c_{j+1}^{in} > 0$. Thus, a forwarding peer receives a non-zero net credit. A larger ω, e.g., $\omega >> \delta$ propagates a larger credit to the peers located on the last hops on the path.

B. Parameters for reciprocity and forgiveness

Reciprocity is parameterized by the threshold for maximum accumulated debt. A larger threshold allows peers to receive multiple chunks before repaying through service. Conversely, a small threshold restricts how many chunks a peer may have to provide to free-riding neighbors without reciprocation. Swarm utilizes a constant threshold parameter set at 400 times the maximum credit for a single chunk ($400 \times \max(c_i^{in})$) across connections. In our model, we employ a default threshold parameter of $1 \times \max(c_i^{in})$, ensuring retrieval of at least one chunk on any connection without settlement. Equation 1 demonstrates $\max(c_i^{in}) = \omega$. We focus on the distribution of tokens transferred after the threshold is reached. The smaller threshold, compared to Swarm, allows us to saturate the threshold also in shorter experiments.

Forgiveness in *Tit4Tok* is determined by the *refresh rate*, quantified in accounting units per second. After Δ seconds, forgiveness extends to $\Delta \times$ *refresh rate* accounting units, capped at the current debt threshold discussed earlier. Swarm sets a *refresh rate* enabling complete debt forgiveness after 20 seconds. This equates to at least 20 chunks forgiven per second on each connection. For more efficient evaluation, we adopt a smaller *refresh rate* set at $1/2 \times \max(c_i^{in}) = \omega/2$ per second.

[1] In his book the acronym ODS refers to open distributed systems while in our paper, it refers to open decentralized systems.

The introduction of reciprocity and limited free service introduces a challenge to fairness. In Section VII-B1, we demonstrate that a constant *refresh rate* distributes the cost of providing limited free service unevenly along retrieval paths. That is because accounting units credited on connections decrease with path length. We designed a different parametrization of reciprocity and limited free service that adapts these parameters based on the distance between the peers adjacent to a connection. Our evaluation shows that this pairwise parametrization can reduce the negative impact the limited free service has on Income-Fairness.

C. When to settle debt with tokens?

When reaching the debt threshold, peers need to decide to either wait for *refresh rate* and rely on the limited free service or to settle debt by transferring tokens. We assume that all peers make use of reciprocity and the limited free service and do not settle debt unless it is required. We also assume that when requesting a chunk without free bandwidth available, a peer will only settle enough debt to request the current chunk.

VII. Evaluation

To evaluate the triad and relevant parameters in our *Tit4Tok* model, we study distance-based credit in isolation and then include reciprocity and limited free service. Finally, we evaluate the effect of additional mechanisms, such as caching and shuffling of connections. We focus on the effect that different parameters have on Income-Fairness and the effectiveness of these mechanisms.

Parameters. The main parameter for distance-based rewards is ω, which we vary in our experiments. As explained in Section VI we bind the *threshold* for maximum debt to ω and the *refresh rate* to $\omega/2$. These settings ensure that at least one chunk can be retrieved for free over any connection every 2 seconds. We use $\omega = 16$ as default parameter, since we find that it most evenly spreads the accounting units from the originator on a single path.

Further, we explore the impact of adjusting the bucket size k, which determines the number of connections maintained in Kademlia. Our experimentation involves a network comprising 10,000 peers. For comparison, on December 12, 2023, the Swarm network contained 9186 active and 8408 staked nodes [4]. We use a storage depth $\delta = 11$. This ensures that on average, four peers are responsible to store a chunk, similar as in the Swarm network. Unless otherwise noted, peer addresses are picked uniformly at random. Thus, the actual number of peers responsible to store a chunk may vary significantly. By default, we set the bucket size to $k = 8$. This results in each peer maintaining 80 connections. This aligns with values in the Swarm network, recently changed from 4 to 20. Connections are used in both directions and chosen uniformly at random.

Workload. We use a uniform distribution of chunks, requesting any address with the same probability. While in real workloads, some files will be more popular than others, even popular files will contain many chunks evenly distributed among peers. Since our chunk addresses are mainly used to determine which peers should be contacted, we believe the assumption of uniform distribution is reasonable. We also did adjust a dataset showing the workload of a public IPFS gateway. In our simulation *originator* nodes represent such gateways. As we saw, the IPFS workload gives similar results as our uniform chunks. We change the number of peers that function as originators varying from 0.5% (50 peers) to 100%. The workload is evenly distributed among the originators. We typically use a workload of 10 million chunks requested over 10 seconds. We use larger workloads where it is necessary to make measurements converge. To mitigate the influence of randomness in our experiments, each value reported is the average over five distinct network graphs.

A. Distance-based credit

In the following, we analyze what effects distance-based credits have on balances and fairness on the accounting layer. Distance-based credits are parametrized by the maximum price ω. We performed extensive simulations, varying the parameter ω and the bucket size k in peers' routing table.

1) Income distribution on the path

We investigate different parameters for ω. A larger ω means that more accounting units need to be sent, but it also changes how these units are distributed among the different peers on a path. We make the following observation:

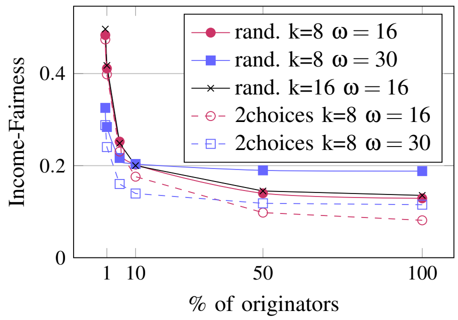
Observation 1: A larger ω parameter gives a smaller fraction of the reward to the first hop in the route.

2) Income-Fairness on the accounting layer

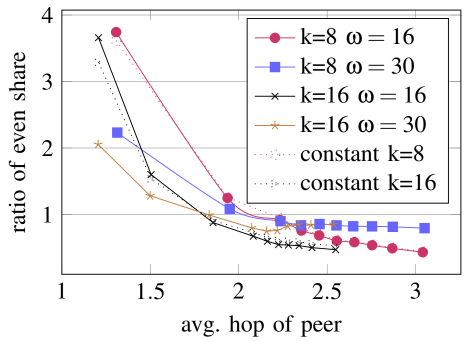
Figure 2a shows the effect of the distribution of request originators (corresponding to gateways in a real life deployment) among peers. Figure 2a shows Income-Fairness for different values of ω. It also shows the difference between networks, where peer addresses are picked uniformly at random, and a network, where every peer receives 2 choices for his address and picks the one with fewer peers within the storage distance. This results in a significantly more even distribution of peer addresses [34]. With 1% or fewer peers as originators, income becomes significantly unequal, especially for smaller ω (see ω =16). With 10% or more peers as originators the more even peer distribution following 2 choices results in lower income fairness. Experiments varying k from 4 to 32 have shown no effect on the Income-Fairness.

With 0.5% originators Income-Fairness is 0.48 with $\omega = 16$ and reduces to 0.33 with $\omega = 30$. Figure 2b investigates the causes for this inequality. Figure 2b shows the fraction of the total accounting units peers receive based on the average hop on which they are located on routes in the experiment. Accounting units are shown as the ratio of even share, where a value of 2 means that a peer receives $2 \times 1/n$ of the total accounting units. The Figure also shows values for a constant reward distribution, where every action is rewarded with a constant value. This also shows the distribution of load in the system.

As can be seen in Figure 2b, with $\omega = 16$, 10% of the peers receive 37% of the total income. The constant distribution shows, that the same 10% of peers also perform 36% of the

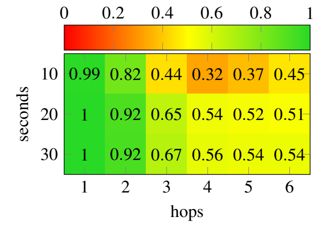


(a) *Income-Fairness* with different number of originators and different ω. Peer addresses are distributed randomly or to one of 2 choices.

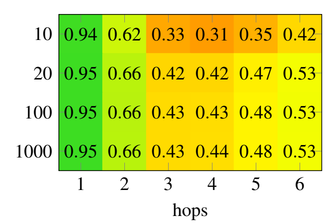


(b) Correlating ratio of even income share receive with average hop of peer, for 0.5% originators. Each dot represents 10% of the peers. Constant variants are included.

Figure 2: Uneven income distribution, due to storage and forwarding discrepancy, and the impact of ω on income faireness



(a) Fraction of paid forwarded chunks with no reciprocity and $\omega = 16$, using a fixed debt threshold and direction.



(b) Fraction of paid forwarded chunks with $\omega = 16$ with reciprocity, allowing to exchange bandwidth for bandwidth.

Figure 3: *Reciprocity and Income-Fairness* The effect of reciprocity on Income-Fairness with 0.5% originators, each requesting 2,000 chunks per second. Heatmaps show the saturation of edges with and without reciprocity.

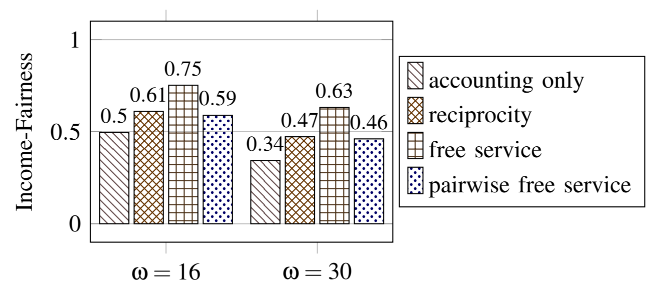


Figure 4: *Limited free service and Income-Fairness* with 0.5% originators. Variants included: accounting only considers only amounts credited on the accouting layer; reciprocity does not provide free service, setting the *refresh rate* to zero; free service uses the *refresh rate* $\omega/2$; pairwise free service, adapts threshold and *refresh rate* to peer distance.

actions. We note that a peer can only be on hop 1, if it has a connection to an originator. With $\omega = 30$ a larger share of the reward is given to the storing action, which is distributed more evenly among peers. This results in better income fairness.

Observation 2: Concentration of few originators results in unequal income, independent of the connectivity. Peers connected to an originator (hop 1) receive more requests and accordingly more income.

Swarm improvement proposal: Observation 2 suggests that Swarm should facilitate and encourage different access modes than through a web gateway (originator) to avoid an uneven load. When uneven load cannot be avoided, a larger ω can still improve Income-Fairness.

B. Reciprocity and Limited free service

This section shows how reciprocity and limited free service effect income fairness at the settlement layer. All measurements use distance-based credit on the accounting layer.

1) Distributing the cost of free service on the path

In Figure 3 we investigate reciprocity without applying the *refresh rate* from the limited free service. Without *refresh rate*, peers still go into debt with each other. The threshold for maximum debt is set to ω. With reciprocity, peers balance out debt given to each other.

To better understand the effect of reciprocity, we also implemented a variant, where peers can go into debt with each other, until the threshold is reached, but the pairwise debts of neighbors are not substracted from each other. We refer to this variant as *no reciprocity*. With no reciprocity, connections hit the threshold significantly faster than with reciprocity, as can be seen from Figure 3a and 3b.

Observation 3: Reciprocity worsens Income-Fairness since requests on the first hop more often produce settlements, then on other hops.

In Figure 4, we investigate the effect of the limited free service on Income-Fairness. We also show results of our *pairwise limited free service*, where threshold and *refresh rate* are set based on the proximity of adjacent peers. We ensure the adjusted threshold is still larger than the accounting units required for any chunk forwarded on that connection.

Figure 4 shows that limited free service can further worsen Income-Fairness, compared to using only reciprocity. Figure 4 shows results for 0.5% gateways. A larger number of gateways gives similar results.

Our pairwise limited free service mitigates this difference, resulting in similar Income-Fairness as the variant not provid-

Table I: Centralization risk through originator cliques

#originators	5	100	5	100
	tok per chunk		internal hops	
random with recip.	11.6	11.1	0.1%	1.6%
clique no recip.	11.0	7.0	5.9%	38.4%
clique with recip.	10.1	6.5	17.5%	45.8%
external clique	9.7	5.9	–	–

ing free service. This shows that our pairwise limits distribute the cost of free service more equally among peers on a path.

Observation 4: The default variant of limited free service significantly reduced Income-Fairness while our pairwise limited free service does not.

Swarm improvement proposal: Following Observation 4, Swarm could achieve a better Income-Fairness by introducing our pairwise limited free service.

2) Centralization risk through gateways

Peers adjacent to gateways receive more income (Observation 2). Accordingly, originators can reduce overall costs by connecting to other originators. Table I shows how clusters of 5 or 100 originators can decrease costs, mainly exchanging chunks among themselves. Reciprocity amplifies this effect as peers prefer bandwidth exchange over token settlements. This incentivizes originators to form clusters, monopolizing traffic. Moreover, gateway operators can gain similar advantages by connecting within the system rather than operating multiple originators or forming *external cliques* through collusion.

Observation 5: Originators are incentivized to form clusters but do not gain significant benefits from external cliques.

3) Caching

Forwarding Kademlia allows peers to cache chunks they forward. We did implement such caching and evaluate its effect on Income-Fairness. Since caching requires a non-uniform workload, we use the adapted IPFS workload for this evaluation. Our experiments show that with few originators caching may increase Income-Fairness from 0.57 to 0.71. However, if the cache at originators is very large, Income-Fairness reduces again (0.67), since mostly unique chunks are requested from the network. With many originators (e.g., 20%) caching can have positive effect, e.g. improving fairness from 0.26 to 0.23. Caching can reduce the imbalance caused by neighborhoods with few peers, reducing the ratio of tokens peers in a neighborhood of size 1 receive from 3.3 to 2.9 times the fair share.

Observation 6: Caching can smoothen imbalances related to the peer distribution in the address space, but increases imbalances with few originators.

C. Income Distribution Across Path

The static routing table in the underlying DHT contributes to unfairness due to a limited number of neighbors. Increasing connections for originators significantly (e.g., $k = 256$) reduces income fairness by 0.2, but maintaining thousands of connections isn't scalable. Instead, we propose a shuffling mechanism where peers periodically change their neighbors to distribute connections over time.

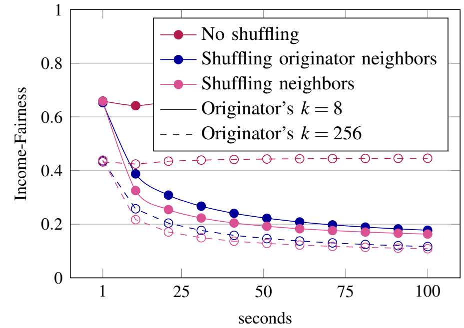


Figure 5: Income-Fairness of different shuffling variants for 0.5% originators with $\omega = 16$, pairwise limited free service and reciprocity enabled.

We evaluate two shuffling variants - only originators shuffling neighbors and all nodes doing so. Figure 5 shows that shuffling may significantly reduce Income-Fairness. Shuffling originators' neighbors also led to a 13.6% reduction in chunk costs since originators may abandon connections with debt.

VIII. Conclusion

We present the *Tit-for-Token* model, which encompasses altruism as limited free service, reciprocity, allowing to exchange service for sercive, and token settlements. We present a comprehensive simulation-based study, to quantify the effect altruism, reciprocity, and different network and workload parameters have on Income-Fairness. Our key findings are:

First, few originators result in bad Income-Fairness, favoring originators' neighbors. With 0.5% originators, 10% of the nodes closest to originators get 36% of requests and income. *Second*, increasing the omega parameter can improve Income-Fairness, when requests are issued by a few originators, e.g., increasing from 16 to 30 improves Income-Fairness with 0.5% originators from 0.5 to 0.34. *Third*, reciprocity and altruism jeopardize Income-Fairness. Originators' neighbors are favored through reciprocity, receiving settlements for up to 96% of requests while other peers only receive settlements for 40-60%. Also, altruism in combination with reciprocity worsens the Income-Fairness, e.g., from 0.61 to 0.75, but our proposed pairwise limit alleviates this issue. *Fourth*, originators' clustering brings the risk of centralization, e.g., with reciprocity, a cluster of 100 originators can reduce the settlement performed by approx. 40% in comparison with a random network layout. *Fifth*, caching can reduce imbalances related to the peer distribution in the address space but increases imbalances with few originators. Sixth, if originators shuffle their connections regularly, Income-Fairness improves significantly, reaching 0.16, a similar level as when all peers originate requests. This approach is also profitable for originators.

Acknowledgements

We thank Filip B. Gotten, Rasmus Øglænd, and Torjus J. Knudsen, who did great work on improving our simulation tool as part of their bachelor thesis. We thank Derouich Abdessalam, who contributed to the shuffling analysis.

REFERENCES

[1] IPFS, 2023. https://ipfs.tech.

[2] libp2p, 2023. https://libp2p.io/.

[3] Swarm, 2023. https://www.ethswarm.org.

[4] Swarm Scan, 2023. https://swarmscan.io/.

[5] Robert Axelrod and William D Hamilton. The evolution of cooperation. *science*, 211(4489):1390–1396, 1981.

[6] Shehar Bano, Alberto Sonnino, Mustafa Al-Bassam, Sarah Azouvi, Patrick McCorry, Sarah Meiklejohn, and George Danezis. Sok: Consensus in the age of blockchains. In *Proceedings of the 1st ACM Conference on Advances in Financial Technologies*, pages 183–198, 2019.

[7] Ingmar Baumgart and Sebastian Mies. S/kademlia: A practicable approach towards secure key-based routing. In *2007 International conference on parallel and distributed systems*, pages 1–8. IEEE, 2007.

[8] James Broberg, Rajkumar Buyya, and Zahir Tari. Metacdn: Harnessing 'storage clouds' for high performance content delivery. *Journal of Network and Computer Applications*, 32(5):1012–1022, 2009.

[9] Andrew Chu, Arjun Arunasalam, Muslum Ozgur Ozmen, and Z Berkay Celik. Behind the tube: Exploitative monetization of content on {YouTube}. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 2171–2188, 2022.

[10] Yang-hua Chu and Hui Zhang. Considering altruism in peer-to-peer internet streaming broadcast. In *Proceedings of the 14th international workshop on Network and operating systems support for digital audio and video*, pages 10–15, 2004.

[11] Bram Cohen. Incentives build robustness in bittorrent. In *Workshop on Economics of Peer-to-Peer systems*, volume 6, pages 68–72. Berkeley, CA, USA, 2003.

[12] Erik Daniel and Florian Tschorsch. Ipfs and friends: A qualitative comparison of next generation peer-to-peer data networks. *IEEE Communications Surveys & Tutorials*, 24(1):31–52, 2022.

[13] Joao FA e Oliveira, Ítalo Cunha, Eliseu C Miguel, Marcus VM Rocha, Alex B Vieira, and Sérgio VA Campos. Can peer-to-peer live streaming systems coexist with free riders? In *IEEE P2P 2013 Proceedings*, pages 1–5. IEEE, 2013.

[14] Marc Eshel, Roger L Haskin, Dean Hildebrand, Manoj Naik, Frank B Schmuck, and Renu Tewari. Panache: A parallel file system cache for global file access. In *FAST*, volume 10, pages 1–14, 2010.

[15] Giulia Fanti, Leonid Kogan, Sewoong Oh, Kathleen Ruan, Pramod Viswanath, and Gerui Wang. Compounding of wealth in proof-of-stake cryptocurrencies. In Ian Goldberg and Tyler Moore, editors, *Financial Cryptography and Data Security - 23rd International Conference, FC 2019, Revised Selected Papers*, LNCS, pages 42–61. Springer, 2019.

[16] Michal Feldman and John Chuang. Overcoming free-riding behavior in peer-to-peer systems. *ACM SIGCOMM Exchanges*, 5(4):41–50, 2005.

[17] Joaquin Delgado Fernandez, Tom Barbereau, and Orestis Papageorgiou. Agent-based model of initial token allocations: Evaluating wealth concentration in fair launches, 2022.

[18] Pawel Garbacki, Dick HJ Epema, and Maarten Van Steen. An amortized tit-for-tat protocol for exchanging bandwidth instead of content in p2p networks. In *First International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2007)*, pages 119–128. IEEE, 2007.

[19] Pedro Garcia Lopez, Alberto Montresor, Dick Epema, Anwitaman Datta, Teruo Higashino, Adriana Iamnitchi, Marinho Barcellos, Pascal Felber, and Etienne Riviere. Edge-centric computing: Vision and challenges, 2015.

[20] Mainak Ghosh, Miles Richardson, Bryan Ford, and Rob Jansen. A torpath to torcoin: Proof-of-bandwidth altcoins for compensating relays. In *Workshop on Hot Topics in Privacy Enhancing Technologies (HotPETs)*, 2014.

[21] Lewis Gudgeon, Pedro Moreno-Sanchez, Stefanie Roos, Patrick McCorry, and Arthur Gervais. Sok: Layer-two blockchain protocols. In Joseph Bonneau and Nadia Heninger, editors, *Financial Cryptography and Data Security*, pages 201–226. Springer, 2020.

[22] Garrett Hardin. Extensions of "The Tragedy of the Commons". *Science*, 280(5364):682–683, 1998.

[23] Cheng Huang, Angela Wang, Jin Li, and Keith W Ross. Understanding hybrid cdn-p2p: why limelight needs its own red swoosh. In *Proceedings of the 18th International Workshop on Network and Operating Systems Support for Digital Audio and Video*, pages 75–80, 2008.

[24] Cornelius Ihle, Dennis Trautwein, Moritz Schubotz, Norman Meuschke, and Bela Gipp. Incentive mechanisms in peer-to-peer networks—a systematic literature review. *ACM Comput. Surv*, 56(1), 2023.

[25] Vikas Jaiman, Leonard Pernice, and Visara Urovi. User incentives for blockchain-based data sharing platforms. *Plos one*, 17(4):e0266624, 2022.

[26] Lian Jian and Jeffrey K. MacKie-Mason. Why share in peer-to-peer networks? In *Proceedings of the 10th International Conference on Electronic Commerce*, ICEC '08, New York, NY, USA, 2008. Association for Computing Machinery.

[27] Seung Jun and Mustaque Ahamad. Incentives in bittorrent induce free riding. In *Proceedings of the 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems*, pages 116–121, 2005.

[28] Henning Kopp, Christoph Bösch, and Frank Kargl. Koppercoin–a distributed file storage with financial incentives. In *Information Security Practice and Experience: 12th International Conference, ISPEC 2016, Zhangjiajie, China, November 16-18, 2016, Proceedings 12*, pages 79–93. Springer, 2016.

[29] Weidong Liu, Dongsheng Peng, Chuang Lin, Zhen Chen, and Jiaxing Song. Enhancing tit-for-tat for incentive in bittorrent networks. *Peer-to-peer networking and applications*, 3:27–35, 2010.

[30] Thomas Locher, Patrick Moore, Stefan Schmid, and Roger Wattenhofer. Free riding in bittorrent is cheap. In *5th Workshop on Hot Topics in Networks (HotNets)*, 2006.

[31] Thomas Locher, Stefan Schmid, and Roger Wattenhofer. Rescuing tit-for-tat with source coding. In *Seventh IEEE International Conference on Peer-to-Peer Computing (P2P 2007)*, pages 3–10. IEEE, 2007.

[32] Petar Maymounkov and David Mazieres. Kademlia: A peer-to-peer information system based on the xor metric. In *International Workshop on Peer-to-Peer Systems*, pages 53–65. Springer, 2002.

[33] Andrew Miller, Ari Juels, Elaine Shi, Bryan Parno, and Jonathan Katz. Permacoin: Repurposing bitcoin work for data preservation. In *2014 IEEE Symposium on Security and Privacy*, pages 475–490. IEEE, 2014.

[34] M. Mitzenmacher. The power of two choices in randomized load balancing. *IEEE Transactions on Parallel and Distributed Systems*, 12(10):1094–1104, 2001.

[35] Bulat Nasrulin, Rowdy Chotkan, and Johan Pouwelse. Sustainable cooperation in peer-to-peer networks. In *2023 IEEE 48th Conference on Local Computer Networks (LCN)*, pages 1–9. IEEE, 2023.

[36] Martin Nowak and Karl Sigmund. A strategy of win-stay, lose-shift that outperforms tit-for-tat in the prisoner's dilemma game. *Nature*, 364(6432):56–58, 1993.

[37] Dongsheng Peng, Weidong Liu, Chuang Lin, Zhen Chen, and Xuehai Peng. Enhancing tit-for-tat strategy to cope with free-riding in unreliable p2p networks. In *2008 Third International Conference on Internet and Web Applications and Services*, pages 336–341. IEEE, 2008.

[38] Michael Piatek, Tomas Isdal, Thomas Anderson, Arvind Krishnamurthy, and Arun Venkataramani. Do incentives build robustness in bittorrent. In *Proc. of NSDI*, volume 7, page 4, 2007.

[39] J. A. Pouwelse, P. Garbacki, J. Wang, A. Bakker, J. Yang, A. Iosup, D. H. J. Epema, M. Reinders, M. R. van Steen, and H. J. Sips. Tribler: a social-based peer-to-peer system. *Concurrency and Computation: Practice and Experience*, 20(2):127–138, 2008.

[40] Bernd Prünster, Alexander Marsalek, and Thomas Zefferer. Total eclipse of the heart–disrupting the {InterPlanetary} file system. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 3735–3752, 2022.

[41] Rameez Rahman, Michel Meulpolder, David Hales, Johan Pouwelse, Dick Epema, and Henk Sips. Improving efficiency and fairness in p2p systems with effort-based incentives. In *2010 IEEE International Conference on Communications*, pages 1–5. IEEE, 2010.

[42] Ashish Rajendra Sai, Jim Buckley, Brian Fitzgerald, and Andrew Le Gear. Taxonomy of centralization in public blockchain systems: A systematic literature review. *Information Processing & Management*, 58(4):102584, 2021.

[43] Ashish Rajendra Sai, Jim Buckley, Brian Fitzgerald, and Andrew Le Gear. Taxonomy of centralization in public blockchain systems: A systematic literature review. *Information Processing & Management*, 58(4):102584, 2021.

[44] Michael Sirivianos, Xiaowei Yang, and Stanislaw Jarecki. Robust and efficient incentives for cooperative content distribution. *IEEE/ACM Transactions on Networking*, 17(6):1766–1779, 2009.

[45] Quinten Stokkink, Alexander Stannat, and Johan Pouwelsee. Foundations of peer-to-peer reputation. In *Proceedings of the 1st International Workshop on Distributed Infrastructure for Common Good*, pages 25–30, 2020.

[46] Ertem Nusret Tas and Dan Boneh. Cryptoeconomic security for data availability committees. In Foteini Baldimtsi and Christian Cachin,

editors, *Financial Cryptography and Data Security*, pages 310–326, Cham, 2024. Springer Nature Switzerland.

[47] Robert L Trivers. The evolution of reciprocal altruism. *The Quarterly review of biology*, 46(1):35–57, 1971.

[48] Viktor Trón. The Book of Swarm - v1.0 pre-release 7 November 17, 2020. Technical report, Neuchatel, Switzerland, 2015.

[49] Viktor Trón, Aron Fischer, Dániel A Nagy, Zsolt Felföldi, and Nick Johnson. Swap, swear and swindle: incentive system for swarm, 2016.

[50] Dimitrios K Vassilakis and Vasilis Vassalos. Modelling real p2p networks: The effect of altruism. In *Seventh IEEE International Conference on Peer-to-Peer Computing (P2P 2007)*, pages 19–26. IEEE, 2007.

[51] Gala Yadgar, Oleg Kolosov, Mehmet Fatih Aktas, and Emina Soljanin. Modeling the edge:{Peer-to-Peer} reincarnated. In *2nd USENIX Workshop on Hot Topics in Edge Computing (HotEdge 19)*, 2019.

[52] Shlomo Yitzhaki and Edna Schechtman. *The Gini methodology: a primer on a statistical methodology*. Springer, 2013.

[53] Manaf Zghaibeh. O-torrent: A fair, robust, and free riding resistant p2p content distribution mechanism. *Peer-to-Peer Networking and Applications*, 11:579–591, 2018.

[54] Manaf Zghaibeh and Fotios C Harmantzis. Revisiting free riding and the tit-for-tat in bittorrent: A measurement study. *Peer-to-Peer Networking and Applications*, 1:162–173, 2008.

[55] Kan Zhang, Nick Antonopoulos, and Zaigham Mahmood. A review of incentive mechanism in peer-to-peer systems. In *2009 First International Conference on Advances in P2P Systems*, pages 45–50. IEEE, 2009.