On the Impacts of Transitive Indirect Reciprocity on P2P Cloud Federations

Eduardo L. Falcão[1], Antônio A. Neto[2], Francisco Brasileiro[1] and Andrey Brito[1]
[1]*Department of Computing and Systems, Federal University of Campina Grande, Campina Grande, Brazil*
[2]*Department of Exact Sciences, Federal University of Paraíba – Campus IV, Rio Tinto, Brazil*

Keywords: Reciprocity Mechanisms, Resource and Availability Asymmetry, Transitive Reciprocity, Cloud Federations.

Abstract: Several P2P systems of resource sharing use cooperation incentive mechanisms to identify and punish free riders, *i.e.*, non-reciprocal individuals. A widespread approach is to use the levels of reciprocity, either directly or indirectly, to decide the extent to which an individual should trust other partners. One restriction of direct reciprocity mechanisms is the inability to foster cooperation between individuals with asymmetrical resources or availability incompatibility. In this work, we evaluate the performance of cloud federations ruled by the combination of the well-known direct reciprocity with transitive reciprocity, a strategy that allows direct reciprocity mechanisms to deal with asymmetry between individuals, while still keeping the benefits of direct reciprocity. For this, we implemented a simulator of resource bartering in cloud federations and experimented it with workloads synthesized from traces of real systems. Our best results showed an average increase of 12.83% and 26.38% on the sharing level of the federation, in an optimistic but unrealistic mechanism setup. When configured in a feasible and realistic manner, the transitive reciprocity was able to increase the sharing level up to an average of 6.02% and 7.53%.

1 INTRODUCTION

One of the challenges with respect to the implementation of a **cloud federation** is to design a fair market model that satisfies the participants and incentivizes cooperation within the community. In general, this resource exchange can be mediated through monetary markets (Dhuria et al., 2017) or in a barter-based approach that is often ruled by the tenets of reciprocity (Haddi and Benchaïba, 2015).

Reciprocity mechanisms are categorized by the way a resource provider assesses the resource requesters, which can be via a direct or indirect approach. In the **direct reciprocity**, a participant uses the knowledge she acquired from her own experiences to assess the past behavior (level of reciprocity) of the requester. In the **indirect reciprocity**, however, third party information can be used for this assessment as well. Both approaches have advantages and disadvantages that make them suitable for particular scenarios. It is reasonable to state that the main advantage of direct reciprocity is the integrity of the information an individual keeps about the behavior of other participants, which prevents *peer collusion* (Ciccarelli and Cigno, 2011). Since it uses private history, direct reciprocity mechanisms are more appropriate for small and mid-sized communities, where repeated interaction among peers is more likely, and also for communities in which the type of resource provided is the same as the one required, such as CPU, GPU and storage – the case of cloud federations. On the other hand, indirect reciprocity mechanisms are commonly used in large-sized P2P systems with wide-ranging resources (*e.g.*, file sharing systems), where the chances of a peer meeting another specific peer and satisfy its requests are rather small.

However, scenarios with *availability asymmetry* (Chuang, 2004) (when two peers can not cooperate because the moments they are consumer and provider, or vice versa, do not match) or with *resource asymmetry* (Feldman et al., 2004) (or conflicts of interests – when two peers can not cooperate because the type of resource supplied by the provider is different from the one required by the consumer) may lead to the creation of credit chains[1] among some participants, and thus some form of indirect reciprocity should be

---

[1] Credit chains are created when, somehow, the resources received by a peer are never reciprocated via direct reciprocity, but may be returned indirectly (through another peer), which yields a cyclic debt.

Falcão, E., Neto, A., Brasileiro, F. and Brito, A.
On the Impacts of Transitive Indirect Reciprocity on P2P Cloud Federations.
DOI: 10.5220/0007686402920299
In *Proceedings of the 9th International Conference on Cloud Computing and Services Science (CLOSER 2019)*, pages 292-299
ISBN: 978-989-758-365-0

taken into consideration in such cases, even if the participants cooperate repeatedly.

In order to deal with this particular source of problem some other works (cf. Section 3) conceived different forms of indirect reciprocity. In this work we leverage the notion of **transitive reciprocity**, presented by Falcão *et al.* (2016), suggested as complement to direct reciprocity. Transitive reciprocity is a more restricted type of indirect reciprocity that can be used in tandem with direct reciprocity mechanisms with the purpose of mitigating the problems (credit chains) arising from availability or resource asymmetry but keeping the advantages of direct reciprocity.

Simulation results (Falcão et al., 2016) suggest that the combination of transitive and direct reciprocity increases the levels of cooperation among the participants when compared to communities in which only direct reciprocity is used. Yet, only scenarios with simple workloads that cover a narrow spectrum of the interesting cases were evaluated. The main results are that, when combined, transitive reciprocity always enhance the performance of direct reciprocity.

However, the workloads experimented were too simplistic and the results may not be representative for real workload scenarios. This study extends such investigation conducted by Falcão (2016), still in simulation lane, through the assessment of the impact of transitive reciprocity in P2P cloud federations, but this time using representative workloads that are synthesized from traces of real systems. In addition, we also investigate the impacts of transitivity in credit chains involving different numbers of peers.

Our main outcomes are the actual performance improvements a cloud federation may experience through transitive reciprocity, and also that in some specific setups, the transitive reciprocity may degrade the overall federation performance (in contrast to results presented earlier). In face of that, this work highlights the main care the member of a cloud federation should take to setup its transitive mechanism properly, to avoid performance degradation.

The rest of this paper is organized as follows. Next section presents the existing reciprocity mechanisms, detailing their main features and drawbacks, to emphasize in what they differ from transitive reciprocity. In Section 3 we show the most relevant related work. The transitive reciprocity is described in Section 4. Section 5 details the reciprocity mechanism used in our simulations, the *Transitive FD-NoF*. In Section 6 we present the simulation model and the description of the evaluated scenario. The simulation results and their corresponding analysis are presented in Section 7. Finally in Section 8 we put forward the main conclusions.

2 RECIPROCITY MECHANISMS

Reciprocity-based incentive mechanisms are strategies that aid providers in prioritizing the most reciprocal consumers for granting its resources. Obviously, this encourages cooperation because free riders (*i.e.*, individuals that consume resources from the community with no compensation) will hardly succeed when there is contention on resources. This decision is made in accordance with the transactions history, which indicates the level of reciprocity of a given peer. Such mechanisms are categorized as *direct reciprocity* or *indirect reciprocity*.

Direct reciprocity takes place on interactions between two individuals – B helps A because A has helped B before, or because B expects A to return this favor in the future (Haddi and Benchaïba, 2015). On the other hand, **indirect reciprocity** takes place on interactions involving at least three individuals, and may happen via *pay-it-forward* (Floyd, 2017) or *reputation-based systems* (Li and Su, 2018). Pay-it-forward reciprocity is a so called altruistic strategy that, in face of free riders, is quite utopic and unfeasible: A donates to B and B returns this favor to C, which is a third party chosen randomly. Note that, in this case, either A, B or C are not expecting something in return. Reputation systems are often used in large-scale P2P systems. In these systems, individuals are encouraged to cooperate in order to render itself as a collaborative and valuable member, and thus be reciprocated by its reputation. Therefore, if A helps B, C will indirectly know that A is cooperative and reward her for her good reputation. The direct reciprocity and pay-it-forward reciprocity can take place via immediate or delayed exchange, whereas reputation systems rely mainly on a delayed exchange since all the information about the reputation of the other peers is collected in the past. Figure 1 illustrates a summary of both types of reciprocity.

The main drawback of reputation systems is that an individual will base their donation decisions on third party information, which may be tampered with by malicious peers to benefit themselves. If on the one hand, the way this information is collected in reputation systems can give room for peer collusion as a disadvantage, on the other hand, the speed in which these pieces of information are diffused by the participants of the system is its main advantage. This feature enables newcomers to collect a reasonable amount of information about the behavior of other participants in a short period of time so that they can properly select whom they should cooperate with. In addition, this speed in which the information is spread enables the community members to swiftly discover the repu-

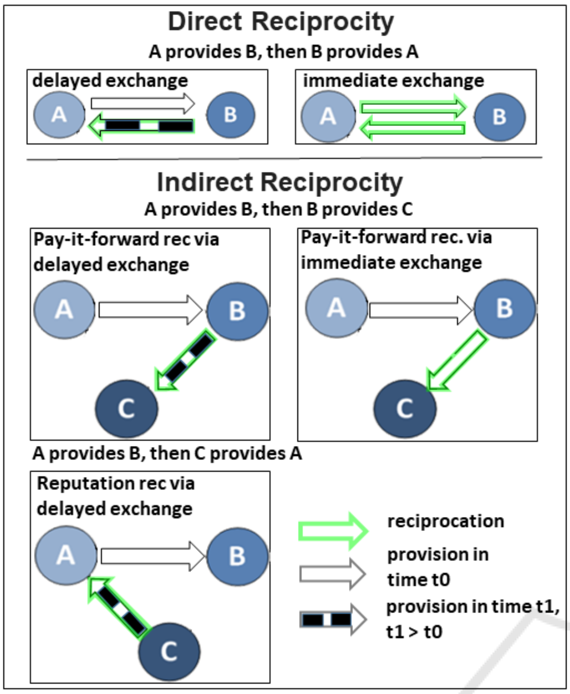

Figure 1: Types of reciprocity mechanisms.

tation of other peers they have never interacted with, what may allow cooperation with unknown peers in a less vulnerable way, as opposed to only interacting with known peers as recommended by the direct reciprocity. Therefore, it is possible to state that indirect reciprocity is also suitable for communities with high levels of resource asymmetry, and not only for communities in which the participants have common interests as required in direct reciprocity — if repeated interaction is not necessary, common interests should not be mandatory after all.

Two closely related problems common to both reciprocity mechanisms are: i) how to treat newcomers and ii) how to mitigate the advantages obtained by whitewashers (individuals that leave and return to the system with a new identity in order to dodge from the consequences of a past in which they acted in a non-cooperative way (Haddi and Benchaïba, 2015)). If the rules to join the system are too strict, the individuals may feel discouraged to join the community, but if, in contrast, these rules are too liberal, the participants would be able to leave the system and appear as a newcomer. Alternatives to this issue could be an entry fee or a proof of work that newcomers should be submitted to be able to join the system (da Costa Cordeiro et al., 2016). Another solution is to disregard negative values for credit/reputation systems (Andrade et al., 2007), thus making no sense for any peer to leave and return to the system since this fact does not change the way other participants value him.

In general, it is possible to state that the main advantage of the **indirect reciprocity** is the quick growth of the database containing information about the behavior of other participants. On the other hand, the main advantage of **direct reciprocity** is the integrity of this same information kept in database. Table 1 presents the characteristics of both mechanisms thus allowing a quick comparison between the main features of each one.

Table 1: Characteristics of direct and indirect reciprocity.

Characteristic	Direct Rec.	Indirect Rec.
database growth	slow	fast
database integrity	guaranteed	not guaranteed
supply and demand	symmetrical	symmetrical and asymmetrical
frequency of interactions	repeated	repeated and occasional
bootstrapping speed	slow	fast
collusion	impossible	possible
identity changing	possible	possible

2.1 The Asymmetry Problem

Direct reciprocity presents the following two restrictions when put into practice individually: time and resources asymmetry. In order to better understand this issue, one must picture a federation consisting of 3 peers (A, B and C) with a conflict of interest, as follows: A has a high processing capacity but insufficient storage, B has enough storage but often needs an extra bandwidth, and C has bandwidth but often needs more processing capacity. This cooperation setup, concerning the types of resource required and offered, leads to the following situation: A always provides to C, B always provides to A, and C always provides to B. In such cases, no resource is directly returned to the providers, and even though all participants involved in this cooperation chain receive what they request, some of them may be seen as free riders.

For explanation purposes, one must consider that peers A, B and C cooperate through a direct reciprocity mechanism in which the balance of B from the perspective of A would be computed by the function

$$\gamma(A,B,t) = \upsilon(B,A,t) - \upsilon(A,B,t),$$

where $\upsilon(A,B,t)$ denotes the total amount of resources A provided to B until time t. Figure 2 depicts the interactions via immediate exchange between three peers with conflicts of interest (the case described on the previous paragraph), and their corresponding balances. It is assumed that $\gamma(A,B,t_0) = \gamma(B,C,t_0) = \gamma(C,A,t_0) = 0$, $t_0 < t_1 < t_2$, and that each peer receives and provides 10 units of resource per time unit.

As shown in Figure 2, as the individuals interact with one another the balances decrease to negative values, what naturally brings distrust. In such cases, it is extremely hard to distinguish between cooperators

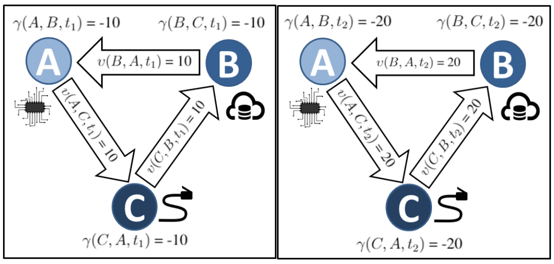

Figure 2: The interests asymmetry problem.

and free riders, since both will have a negative balance for returning no favors, either intentionally (free riders) or unintentionally (asymmetry).

Next section depicts how some related work address this issue.

3 RELATED WORK

The main indirect reciprocity mechanisms typically use reputation systems, transitive credit systems, or the pay-it-forward principle.

Regarding the cryptography technique, there exists *Dandelion* (Sirivianos et al., 2009) and *T-Chain* (Shin et al., 2017), based respectively on a global credit system with tit-for-tat and on pay-it-forward, both thought for file sharing systems. The cryptography applied on Dandelion and T-Chain is efficient in promoting cooperation even in the face of free riders. However, this idea is only feasible for file sharing systems, since files are encryptable.

Apart from the cryptography strategy, which clearly is not applicable to cloud federations, we may also cite, still in the context of file sharing, the *Cluster Based Incentive Mechanism* (CBIM) (Zhang and Antonopoulos, 2013), the *K-Cycle* (Eidenbenz et al., 2012) and Menasché *et. al.* (2010). CBIM and K-Cycle organize peers with asymmetric resources in what the authors call *barter rings* or *barter cycles*, forcing the peers comprising the ring to cooperate before consuming. Menasché *et. al.* (2010) suggest that cycles of indirect reciprocity cooperation should be converted into direct reciprocity.

When it comes to mechanisms that could be straightforwardly applied to P2P cloud federations and computational grids, but without relying indiscriminately on third-parties (such as the transitive reciprocity we rely on), there are the following relevant work: *CompactPSH* (Bocek et al., 2009), *PledgeRoute* (Landa et al., 2009) and *Scrivener* (Nandi et al., 2005). In these mechanisms, indirect reciprocity is broken down into two steps: *credit transfer* and direct reciprocity. The idea is to strengthen the economy of the system since the transfer of credits allows interactions that would not happen between peers with asymmetric resources.

4 TRANSITIVE RECIPROCITY

Transitive reciprocity is a more limited form of indirect reciprocity, which is designed to be put into practice in tandem with direct reciprocity but while still keeping the integrity of database with transactions history (which is not the case for reputation systems). For instance, when A requests resources from B, and somehow the latter can not meet A's request, B can ask other peers that are in debt with him (*e.g.*, C) to do A this favor as a kind of repayment to B. Further, once A receives the resources, she must be informed by C that that favor was done under B's request. Also, A will increase the balance of B in proportion to the time period and amount of resources provided by C, but obviously A will not credit C and C will not debt A.

With the aid of transitive reciprocity, the distrust raised by the asymmetry problem (Fig. 2) would be solved. The decrease of balances that took place on each time period due to the cooperation chain ($A \rightarrow C \rightarrow B \rightarrow A$) would give place, in face of transitive reciprocity, to a balance stability. At the end of t_1, $\gamma(A,B,t_1) = \gamma(B,C,t_1) = \gamma(C,A,t_1) = 0$, as well as at the end of t_2, $\gamma(A,B,t_2) = \gamma(B,C,t_2) = \gamma(C,A,t_2) = 0$, instead of -10 and -20 for t_1 and t_2 respectively, since for each time period the resource provision would be indirectly reciprocated and all the debts would be paid.

From the related work we may note that the main differences are the protocol to enable cooperation among peers with asymmetry, and the application context, which may require special treatment. The change of the protocol may not entail significant changes in the results of the performed simulations, since the protocols are only enablers of the same interaction through different approaches/idea of indirect reciprocity. However, transitive reciprocity is more efficient in terms of amount of exchanged message – to allow interaction through transitive reciprocity between three peers, its protocol only needs 5 messages, while Scrivener and CompactPSH needs respectively 6 and 10 messages (cf. (Nandi et al., 2005; Bocek et al., 2009)).

5 THE TRANSITIVE FD-NoF

In the Fairness-Driven Network of Favors (or FD-NoF, for short), the most reciprocal participants are prioritized for the concession of favors through a system of balances that, considering functions defined in Section 2.1, are calculated by the following equation:

$$\gamma(A,B,t) = max\{0, \upsilon(B,A,t) - \upsilon(A,B,t) + \log \upsilon(B,A,t)\}.$$

By preventing the balances from being negative, this definition does not allow malicious peers to manipulate their balances by changing their identity to pretend to be a newcomer. Furthermore, it takes into consideration an amortized and historical portion of donations that *A* has received from *B* in the past. This enables *A* to distinguish a peer *C* that has never provided resources from a cooperative peer *B* that has provided to *A* in the past but has received from *A* at least the same amount of resources provided.

In order to assure *fairness*, the ratio between the total amount of resources received and the total amount of resources provided up to a particular point in time, the FD-NoF suggests that each participant should manage the amount of resources offered to other peers of the federation. Each participant is provided with a feedback control loop mechanism and must define a minimum (τ_{min}) and maximum threshold (τ_{max}), thus determining an interval $[\tau_{min}, \tau_{max}]$ that represents the desired levels of fairness. One must assume that the amount of resources that *A* offers to any other peer *B* in time *t* is expressed by the function $\alpha(A,B,t)$. If at any moment the fairness of *A* in relation to a peer *B* is lower than τ_{min}, the controller will continuously subtract a fixed value Δ from $\alpha(A,B,t)$ over the subsequent periods of time until the fairness of *B* under the perspective of *A* gets higher than τ_{min}. On the other hand, if the fairness of *A* in relation to *B* is superior to τ_{max}, the controller will add Δ to the value of $\alpha(A,B,t)$ over the subsequent periods of time until the fairness of *A* in relation to *B* assumes an inferior value to τ_{max} — at this stage, once the desired level of fairness is reached, *A* will try to have an accumulated balance in the perspective of this participant. Finally, when the fairness of *A* in relation to *B* is within the interval $[\tau_{min}, \tau_{max}]$, the controller will run a Hill Climbing algorithm that uses the latest values of fairness to decide whether there should be an increase or decrease in $\alpha(A,B,t+1)$ to maximize the fairness of *B* perceived by *A* within the interval $[\tau_{min}, \tau_{max}]$.

This brief discussion aims to make this work self-contained with respect to the the FD-NoF mechanism. Additional information about the algorithm and its performance can be found in (Falcão et al., 2016).

The goal of transitive reciprocity is to enable cooperation between participants that show any type of asymmetry. Therefore, whenever a peer has idle resources and the FD-NoF mechanism suggests this peer should not share them, the transitive reciprocity must be put into practice to verify if there is any credit chain between the requester and the provider, what could enable cooperation through transitivity. Each peer is free to setup its own mechanism autonomously by choosing the maximum allowed length of chain ($\chi \in \mathbb{N}_{\geq 3}$) and also the percentage of peers considered for trying cooperation at each level of the chain, $\eta \in [0,1]$. Obviously, the higher are χ and η the higher will be the probability of cooperation, but also more congested will be the federation network.

The FD-NoF mechanism with this additional enabler is what we call the **Transitive FD-NoF**.

6 EVALUATION OF THE TRANSITIVE FD-NoF

In order to assess the performance of the Transitive FD-NoF in P2P cloud federations, we advanced the simulator built by Falcão *et. al.* (2016) to allow the experimentation of realistic workloads, and also to enable cooperation in transitive chains with more than three peers. The simulation model with these additional features is described next.

6.1 Simulation Model

The federation $\mathbb{F}$ consists of a community comprised by $\mathcal{N}$ peers. The simulation proceeds in discrete steps and, therefore, the balance ($\gamma : \mathbb{F} \times \mathbb{F} \times \mathbb{N} \to \mathbb{R}_{\geq 0}$) and quota ($\alpha : \mathbb{F} \times \mathbb{F} \times \mathbb{N} \to \mathbb{R}_{\geq 0}$) functions are best described in the form of step functions.

Assume that each peer has a total resource capacity of $\mathcal{C} \in \mathbb{R}_{\geq 0}$. A peer in consumer state at any step *t* will have a total resource demand $\mathcal{D}_{total} = \mathcal{D} + \mathcal{C}$, of which $\mathcal{C}$ units are provided by the local resources and $\mathcal{D}$ units are requested to the federation. A peer is in provider state when $\mathcal{D}_{total} < \mathcal{C}$, which means that it can provide up to $\mathcal{C}$ resource units to the federation. For two-party interactions, a peer *A* in a provider state at time step *t* may provide up to $\alpha(A,B,t)$ units of resources to any peer *B* in consumer state at step *t*.

When cooperating through transitive reciprocity and the number of peers involved in the credit chain is three ($\chi = 3$), the amount of resources provided is computed as follows. A peer *C* in a provider state at step *t* may provide to a peer *A* in a consumer state on behalf of a peer *B* up to the minimum between $\alpha(C,B,t)$ (the amount that *C* would provide directly to *B*) and $\alpha(B,A,t)$ (the amount that *B* would provide to *A*, since *B* would not require *C* to provide more resources to *A* than *B* itself would provide). The same logic is applied for $\chi > 3$. Roughly, this means that the peers involved in the transitive reciprocity never donate more than the limit imposed by the controller.

Whenever a peer is in a consumer state, first she will send requests to participants she knows she has credit with, in a direct reciprocity basis. If none of them can meet her request, then she will send requests to peers that could meet requests via transitive reciprocity, with the aid of any credit chain. Let us call these intermediate individuals the *transitive peers*. Then, each transitive peer, which will denote one level of the chain, should forward this request only to a portion (η) of the participants of the federation. Finally, if none of these requests are met, the consumer sends direct requests to unknown peers, randomly chosen, for establishing new links. At each time step, upon receiving the requests from consumer peers, the simulator randomly selects a peer in a provider state, and this peer prioritizes the granting of resources to consumer peers according to the protocol established by the NoF — peers with higher balances are prioritized, and if there is more than one requester with the same balance, the provider distributes its idle resources equanimously among them.

In simulations in which $\chi > 3$, the consumer will always try to find the credit chain with lowest length possible to the provider. In other words, if for instance $\chi = 5$, the consumer will first try to find a provider through a chain with a single level ($\chi = 3$), and in case it is not possible, she will increment χ until she finds a chain to a provider. This process is repeated until her demand has been fully met or until the requester has explored chains of length $\chi = 5$ and no credit chain has been found.

The source code of the Transitive FD-NoF simulator is available at https://github.com/antonionetto20/Transitive-FD-NoF.git.

6.2 Workload Synthesized from Real Traces

In terms of client applications, cloud federations may have similar applicability to that of P2P opportunistic computational grids, suitable for fault tolerant applications such as *Bag-of-Tasks* (BoT). Therefore, we used a workload generator presented by Carvalho and Brasileiro (2012), a software that uses traces from real systems, provided by the Grid Workload Archive at http://gwa.ewi.tudelft.nl/, in order to synthesize realistic workloads based on parameters such as number of peers, users, maximum duration of a job and duration of the whole workload.

With the aid of this software, we generated a workload of 72 hours, comprising a federation of 60 organizations (peers) and 10 users per organization.

Since the simulator models the exchange of resources in discrete turns, it was necessary to map the jobs generated by the workload to the turns of the simulation. This was done considering *time grains*, *i.e.*, mapping all jobs that start and/or finish in a given time slot for one turn of the simulation. In our simulations, to approximate the duration of a turn to the jobs running time, the grain was defined as the average of the total duration of all jobs: 942 seconds. The factors are the reciprocity mechanism (direct or transitive), the total resource capacity of each peer ($\mathcal{C}$) and, if the transitivity is enabled, the maximum length of the credit chain (χ) and the percentage of peers considered at each level of the chain (η) – recall that the peers' demands comes from the workload generator. We varied the resource capacity of the peers to understand the impact of the transitivity in scenarios with different levels of resource contention. In addition, we also changed χ and η to analyze the trade-off between the cost of increasing the number of peers involved in the transitive cooperation (which would increase the amount of messages sent, possibly congesting the network) and its benefits (which would be the increase in the amount of shared resources thanks to the number of transitive peers).

Once the difference in performance between the FD-NoF and the Transitive FD-NoF in a turn tends to be marginal, the evaluation metric is computed by the total amount of resources donated in the Transitive FD-NoF minus the total amount of resources donated in the FD-NoF, in all turns of the simulation. With this value we can measure the **percentage of increase in donation** brought by the transitive reciprocity.

In all simulations, the FD-NoF mechanism is set with $\Delta = 0.05 \cdot \mathcal{C}$, $\tau_{min} = 0.75$ and $\tau_{max} = 0.95$, values suggested by Falcão *et. al.* (2016) for performing well in wide range of scenarios.

7 RESULTS AND DISCUSSION

Figure 3 presents the percentage of increase in total donation when comparing the same scenarios ran with Transitive FD-NoF and the FD-NoF. We simulated scenarios in which all the peers are set only with $\mathcal{C} = 50$ and scenarios in which they are all set with $\mathcal{C} = 100$. In addition, we also changed the maximum chain length when the transitive reciprocity is enabled between $\chi \in \{3,4,5\}$. We first explore transitivity to its edge — the percentage of peers tried on each chain level is 100% ($\eta = 1$). Each box-plot presents the distribution of the **percentage of increase in donation** in 30 replications, where the seed for generating random numbers is modified, which affects the order in which providers and consumers are selected by the simulator for the provision of resources. The confidence intervals for these distributions are plot with a 95% confidence level.

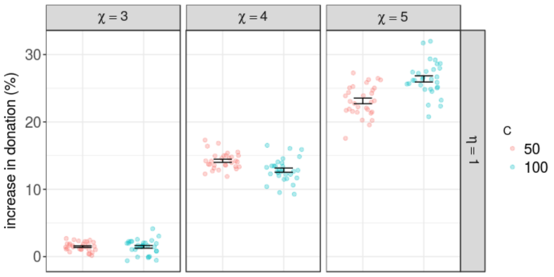

Figure 3: Distribution and confidence intervals of percentage increase in resource donation provided by transitivity in scenarios with $\chi \in \{3,4,5\}$, $\mathcal{C} \in \{50,100\}$ and $\eta = 1$.

From Figure 3 one may observe that the transitive reciprocity increased the percentage of donation in vast majority of the scenarios simulated with $\eta = 1$, specifically, 97.2%. The remaining scenarios, configured with $\mathcal{C} = 100$ and $\chi = 3$, showed performance degradation as can be seen on Figure 3 (a few blue points below zero). This fact was not perceived before in (Falcão et al., 2016), perhaps because it was used a very simple workload that didn't generate cooperation scenarios with higher complexity. This degradation may be experienced when specific interactions, that occur at a given moment, depend on the occurrence of previous interactions, since the existence (or not) of a past interaction that happened through the presence (or absence) of the transitive reciprocity may alter the disposition of balances and quotas between the peers, in such a way that prevents certain donations. However, when the length of the transitive chain (χ) is increased, augmenting the chances of transitive cooperation, there is a considerable increase in the amount of resources shared. For scenarios in which the peers have $\mathcal{C} = 100$, the average increase in donation was 12.83% and 26.38% for $\chi \in \{4,5\}$. Finally, we observed that the increase in donation brought by the transitive reciprocity in scenarios with different resource contention (different $\mathcal{C}$) may differ significantly for $\chi \in \{4,5\}$, although with no clear pattern since for $\chi = 4$, $\mathcal{C} = 100$ underperformed scenarios with $\mathcal{C} = 50$ but for $\chi = 5$, $\mathcal{C} = 100$ overperformed $\mathcal{C} = 50$.

However, exploiting all possibilities of transitive chains between the provider and the consumer may not be realistic depending on $\mathcal{N}$, χ, and η. Scenarios with $\mathcal{N} = 60$, $\chi = 5$ and $\eta = 1$ allow a number of combinations up to $59 + (59 \cdot 58) + (59 \cdot 58 \cdot 57) = 198530$ between a single provider and all the other participants. Considering all the providers at a given time, the amount of messages sent would surely congestion the network, besides the delay on decision of each intermediate peer on whether it should cooperate or not.

For this reason, we also evaluated scenarios in which the transitive reciprocity considered a reduced amount of participants at each level of the chain, $\eta \in \{\frac{1}{6}, \frac{1}{3}\}$. The increase in donation of such simulations are presented in the box-plots of the Figure 4.

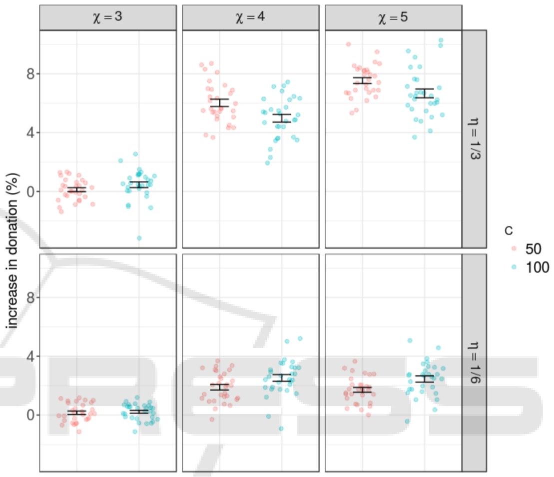

Figure 4: Percentage increase in resource donation provided by transitive reciprocity in scenarios with $\chi \in \{3,4,5\}$, $\mathcal{C} \in \{50,100\}$ and $\eta \in \{\frac{1}{6}, \frac{1}{3}\}$.

In scenarios setup with $\chi = 3$ and $\eta \in \{\frac{1}{6}, \frac{1}{3}\}$ there were some scenarios with performance degradation due to the low value of η. Further, we may also note that when $\chi = 3$, there is no meaningful difference on increasing η from $\frac{1}{6}$ to $\frac{1}{3}$. However, when $\chi \in \{4,5\}$ the amount of peers considered for cooperation increases substantially, and then increasing η incurred in a higher level of resource provision. Scenarios set with $\chi \in \{4,5\}$ and $\eta = \frac{1}{3}$, for instance, had solely performance improvements, 6.02% and 7.53%, on average. Therefore, $\chi = 4$ and $\eta = \frac{1}{3}$ can be considered and interesting setup for presenting a good cost-benefit, at least for the workload experimented, since to explore all cooperation possibilities a consumer would only need to send 420 messages.

8 CONCLUSIONS

This work evaluates the impact of transitive reciprocity in P2P cloud federations with workload synthesized from traces of real systems. To this goal, a simplified simulation model was conceived to allow the investigation of the performance of transitive reciprocity in different scenarios. To the best of our knowledge, only the present work evaluates the impacts of the chain length and the amount of peers considered for cooperation on each level of the chain.

Simulation results showed that in some scenarios (2.8%) in which all interactions are tried ($\eta = 1$) transitive reciprocity may degrade the federation overall performance – the amount of shared resources is decreased, a result not seen in any related work. However, our main findings are that transitive reciprocity can actually increase the sharing level in a more realistic scenario in which less transitive combinations between consumer and provider are considered — Transitive FD-NoF increased the percentage donation in 6.02% and 7.53% in scenarios setup with $\eta = \frac{1}{3}$ and $\chi \in \{4, 5\}$, respectively.

ACKNOWLEDGMENT

This research was partially funded by the EU-BRA SecureCloud project (EC, MCTIC/RNP, and SERI, 3rd Coordinated Call, H2020 Grant agreement no. 690111) and by the EU-BRA ATMOSPHERE project (EC and MCTIC/RNP, 4th Coordinated Call, H2020 Grant agreement no. 777154).

REFERENCES

Andrade, N., Brasileiro, F., Cirne, W., and Mowbray, M. (2007). Automatic grid assembly by promoting collaboration in peer-to-peer grids. *Journal of Parallel and Distributed Computing*, 67(8):957 – 966.

Bocek, T., Hecht, F. V., Hausheer, D., Stiller, B., and Elkhatib, Y. (2009). Compactpsh: An efficient transitive tft incentive scheme for peer-to-peer networks. In *2009 IEEE 34th Conf. on Local Computer Networks*, pages 483–490.

Carvalho, M. and Brasileiro, F. (2012). A user-based model of grid computing workloads. In *2012 ACM/IEEE 13th International Conference on Grid Computing*, pages 40–48.

Chuang, J. (2004). Designing incentive mechanisms for peer-to-peer systems. In *1st IEEE International Workshop on Grid Economics and Business Models, 2004. GECON 2004.*, pages 67–81.

Ciccarelli, G. and Cigno, R. L. (2011). Collusion in peer-to-peer systems. *Computer Networks*, 55(15):3517 – 3532.

da Costa Cordeiro, W. L., Santos, F. R., Barcellos, M. P., Gaspary, L. P., Kavalionak, H., Guerrieri, A., and Montresor, A. (2016). Making puzzles green and useful for adaptive identity management in large-scale distributed systems. *Computer Networks*, 95:97 – 114.

Dhuria, S., Gupta, A., and Singla, R. (2017). Resource pricing in cloud federation: A review. *International Journal of Innovations & Advancement in Computer Science*, 6(12).

Eidenbenz, R., Locher, T., Schmid, S., and Wattenhofer, R. (2012). Boosting market liquidity of peer-to-peer systems through cyclic trading. In *2012 IEEE 12th International Conference on Peer-to-Peer Computing (P2P)*, pages 155–166.

Falcão, E. L., Brasileiro, F., Brito, A., and Vivas, J. L. (2016). Enhancing p2p cooperation through transitive indirect reciprocity. In *2016 IEEE 36th International Conference on Distributed Computing Systems*.

Falcão, E., Brasileiro, F., Brito, A., and Vivas, J. (2016). Enhancing fairness in p2p cloud federations. *Computers & Electrical Engineering*, 56:884 – 897.

Feldman, M., Lai, K., Stoica, I., and Chuang, J. (2004). Robust incentive techniques for peer-to-peer networks. In *Proc. of the 5th ACM Conf. on Electronic Commerce*, pages 102–111, New York, NY, USA. ACM.

Floyd, R. E. (2017). Pay it forward. *IEEE Potentials*, 36(2):5–47.

Haddi, F. L. and Benchaïba, M. (2015). A survey of incentive mechanisms in static and mobile {P2P} systems. *Journal of Network and Computer Applications*, 58:108 – 118.

Landa, R., Griffin, D., Clegg, R. G., Mykoniati, E., and Rio, M. (2009). A sybilproof indirect reciprocity mechanism for peer-to-peer networks. In *INFOCOM 2009, IEEE*, pages 343–351.

Li, S. and Su, W. (2018). The research of reputation incentive mechanism of p2p network file sharing system. *International Journal of Information and Computer Security*, 10(2-3):149–169.

Menasché, D. S., Massoulié, L., and Towsley, D. (2010). Reciprocity and barter in peer-to-peer systems. In *Proc. of the 29th Conf. on Information Communications*, pages 1505–1513, NJ, USA. IEEE Press.

Nandi, A., Ngan, T.-W., Singh, A., Druschel, P., and Wallach, D. S. (2005). Scrivener: Providing incentives in cooperative content distribution systems. In *Middleware*, volume 3790 of *Lecture Notes in Computer Science*, pages 270–291. Springer.

Shin, K., Joe-Wong, C., Ha, S., Yi, Y., Rhee, I., and Reeves, D. S. (2017). T-chain: A general incentive scheme for cooperative computing. *IEEE/ACM Transactions on Networking*, 25(4):2122–2137.

Sirivianos, M., Yang, X., and Jarecki, S. (2009). Robust and efficient incentives for cooperative content distribution. *IEEE/ACM Trans. Netw.*, 17(6):1766–1779.

Zhang, K. and Antonopoulos, N. (2013). A novel bartering exchange ring based incentive mechanism for peer-to-peer systems. *Future Generation Computer Systems*, 29(1):361 – 369.