Ballester Lafuente and Seigneur *Journal of Trust Management* (2015) 2:7
DOI 10.1186/s40493-015-0018-0

Journal of Trust Management
a SpringerOpen Journal

RESEARCH Open Access

Extending trust management with cooperation incentives: a fully decentralized framework for user-centric network environments

Carlos Ballester Lafuente* and Jean-Marc Seigneur

* Correspondence:
carlos.ballester@unige.ch
University of Geneva, ISI/ISS & Medi@LAB, GSEM & SdS, CUI 7 Route de Drize, Carouge, CH 1227, Switzerland

Abstract

While trust management systems can be used in isolation in order to provide robustness to a given architecture, cooperation incentives can be used to complement and collaborate with trust management systems as users can benefit from them while using the system, thus encouraging user's good behaviour. We have designed a fully decentralized trust management and cooperation incentives framework for user-centric network environments composed by three main components, the *identity manager*, the *trust manager* and the *cooperation manager*. In this article, we present how we integrate our trust management and cooperation incentives framework with a collaborative wireless access sharing service, being the aim of the article to evaluate its feasibility from a bootstrapping and survivability point of view. Our results obtained through simulation prove that the values for bootstrapping and data depletion times are well inside acceptable ranges, given that the total user base for the framework in the world is big enough while using friend-of-a-friend chains.

Keywords: Wi-Fi; Collaborative sharing; Trust; Cooperation incentives; Trust points

Introduction

According to the International Telecommunication Union (ITU) [1], the number of subscribers using mobile broadband Internet services has raised from 268 million in 2007 to an impressive 2.1 billion users in 2013, accounting for more than the 50 % of the world's Internet usage.

Wireless networks today are partially being formed by nodes (e.g. Internet access points, smartphones, femtocells) that are owned and carried by humans. As such, these User-centric network architectures (UCNs) are giving rise to new Internet architectures, where broadband access is complemented by e.g. Wireless Fidelity (Wi-Fi) clouds, having a strong involvement of the Internet end-user. This represents a paradigm shift in the Internet evolution, as the user may be in control of parts of the network, in a way that is acknowledged (or not) by Internet stakeholders. In such scenarios where several strangers are expected to interact for the sake of robust data transmission, trust and cooperation incentives are of vital importance as these establish a way for the nodes involved in the system to communicate with each other in a safe manner, to share services and information, and above all, to form communities that assist in sustaining robust connectivity models. Without cooperation incentives, and following the principle of the tragedy of the

commons, users would always do what's better for them and behave selfishly, thus not cooperating for the sake of data transmission but for their own benefit.

While trust management systems can be used in isolation in order to provide robustness to a given architecture, cooperation incentives can be used to complement and collaborate with trust management systems as users can benefit from them while using the system, thus encouraging user's good behaviour. All in all, the relation between trust and cooperation incentives can greatly help into building a really solid and reliable architecture, being this the main purpose and scope of this article.

We have designed a fully decentralized trust management and cooperation incentives framework for user-centric network environments. Our proposed framework is composed by three main components, the *identity manager*, the *trust manager* and the *cooperation manager*. The identity manager should be able to deal with multiple crypto-ids per user as detailed below. The trust manager is composed by a dispositional trust adaptation module and by a proven Sybil [2] attack resistant trust metric taken from the work of Seigneur [3] and, finally, the cooperation manager should provide the right incentives such as points or a virtual currency, which can be exchanged for services and can be gained when providing them, to help trust management to achieve an overall more secure and reliable architecture.

In this article, we present how we integrate our trust management and cooperation incentives framework with a collaborative wireless access sharing service, being the aim of the article to evaluate its feasibility from a bootstrapping and survivability point of view. Our results obtained through simulation prove that the values for bootstrapping and data depletion times are well inside acceptable ranges, given that the total user base for the framework in the world is big enough while using friend-of-a-friend (FOAF) [4] chains. This allows trust points to be transferred or lent from one entity to another along the chain thus providing strong incentives for cooperation, reassuring the effectiveness of our system.

The rest of the document is organized as follows. First, section 2 presents the current state of the art. Following, section 3 describes our framework model and design. After, section 4 the methods used for simulation and validation. Next, section 5 shows the simulation and the results obtained from it. Finally, section 6 concludes the paper.

Background and literature review

There has been a lot of work and research already done in the fields of trust management and reputation, cooperation incentives and survivability of decentralized user-centric networks. The aim of this section is to present the most relevant work already done and related to this article.

This review will allow us to identify the commonalities already addressed by existing frameworks, identify their shortcomings and pitfalls and identify the gap present in these fields in order to be able to better compare them to our own framework.

Trust management

In the work presented in the thesis of S. Ries [5], the author provides an approach based on trust management in order to improve the selection of reliable interaction partners. The main goal of his approach is to estimate the trustworthiness of a given entity with the highest possible accuracy to improve the average quality of the interactions with it. The trustworthiness of an agent or entity is derived from the evidence

collected during past interactions. In order to achieve that, current Bayesian trust models are extended and improved in several aspects, including a better integration of recommendations by third parties.

This aspect is crucial as there are many scenarios where direct evidence between entities is unattainable or scarce. This situation, where a system or architecture is either very new or hasn't been used much, thus not being able to provide meaningful information to its participants is knows as the *cold-start problem*. For example, such a scenario can be found in a new recommender system where almost all the participants are new and nodes have seldom interacted with each other, thus not being able to form direct opinions from direct interactions with other nodes.

The proposed approach provides a solution involving the robust integration of recommendations provided by third parties, especially considering possible attacks by entities providing on purpose misleading recommendations, either individually or collectively. The approach is validated through simulation, showing results over a set of 15 populations, which have been canonically derived from the system model, modelling entities with different typical behaviours. Furthermore, the results obtained by simulation regarding collaboration between agents in an opportunistic network prove that the model provides high accuracy about the estimation of an entity's trustworthiness and the average quality of interactions to find the best interaction partner.

Martucci et al. [6] propose an identity management model which supports role-based pseudonyms, which are different digital identities per user based on the role or context the user is operating in at a given point of time, and which can support the use of trust and reputation systems while still providing a reasonable amount of privacy protection and anonymity and at the same time avoiding Sybil attacks. Users' privacy protection requires actions that cannot be linked one another, implying that an external observer is not able to link two actions or their outcomes belonging to the same user. The problem they face is that building up trust and reputation usually requires long-term identifiers that can be in fact linked over several transactions for a given agent or entity. To tackle this problem they propose an architecture to generate pseudonyms based on roles, which in turn are bound to a given set of services called a service context. Their proposal offers unique long-term identifiers which are the basis for trust and reputation systems, allowing to build behaviour histories about the other entities in the system, while still providing unlinkability between the actions performed by a given entity in different service contexts and detecting Sybil identifiers to avoid whitewashing, badmouthing and other Sybil related attacks. In order to achieve Sybil-free pseudonyms, they use a cryptographic construction which generates one self-signed pseudonym for each of the contexts the user has to interact with, all derived from an initial identifier provided by a trusted third-party on the bootstrap step (*hence, it is not a fully decentralized system*). Contexts are directly created by the service providers present in the system, and they contain a unique hashed value. By combining this hashed value with a newly auto-generated public key and their initial identifier, the agents can generate context-specific pseudonyms to interact with that given context. To conclude, they demonstrate that it is still possible to detect if a single agent has created more than one pseudonym for a unique context by means of cryptographic calculations, hence effectively avoiding the possibility of Sybil attacks while still preserving the privacy of the entity as actions carried out in different contexts cannot be linked.

For Ziegler et al. [7], besides understanding the information and relations in between entities, knowing about their credibility is equally important and crucial, and thus trust and trust metrics are needed to evaluate trust relationships between individuals. One of their main contributions to the field is an extensive trust metric classification dividing trust metrics into two big groups, namely a global one, which takes into account all peers and trust links connecting them in the whole system and a local one, which takes into account personal bias and compute a more personalized trust, further subdividing these categories into more specific ones. The second contribution the authors make in their work is *Appleseed*, a trust metric designed for a Semantic Web scenario and which is based on spreading activation strategies. Appleseed works with partial trust graph information where nodes are queried only when needed, and where nodes make their manual trust values publicly available, thus posing a threat to privacy. Finally, the authors compare their trust metric with Advogato [8], and evaluate its attack resistance. Advogato is a trust metric that evaluates a set of peer certificates in order to be able to accept new user accounts, where the certificates are represented as a graph, with each account as a node and each certificate as an edge, in order to accept as many valid accounts as possible while reducing the impact of attackers.

In EigenTrust [9], the global reputation of each peer is given by the local trust values assigned to it by other peers, weighted by the global reputations of the assigning peers. Trust values are normalized so no peer can assign arbitrarily high or low values to other peers in order to subvert the system. Then trust is aggregated by taking into account peers' recommendations weighted with the trust the node has on those other peers. Each peer has a number M of score managers and since each peer also acts as a score manager, it has assigned a set of daughters referenced by the indexes of peers whose trust value computation is covered by the peer. As a score manager, a peer also maintains the opinion vector of its daughter peers and it also learns the set of peers, which downloaded files from its daughter peers, receiving trust assessments from these peers referring to its daughter peer. Finally, a peer also gets to know the set of peers which its daughter peers downloaded files from and the trust assessments on those peers from its daughter peers. Their results have shown a reduced number of inauthentic files on the network under a variety of threat scenarios. Furthermore, rewarding highly reputable peers with better quality of service incentivizes honest peers to share more files and to self-police their own file repository for inauthentic files.

Cooperation incentives

In Feldman et al. [10], the authors focus on the issues present in peer-to-peer (P2P) networks and that make the challenge of achieving cooperation more complicated than in other environments. Some of those issues are large populations, self-interest, zero-cost identities, dynamicity of the system and short-lived population. In order to address cooperation and to provide incentives, they have created a reciprocative decision function with the following three requirements:

1. Can use shared and subjective history
2. Can deal with defections
3. Is robust against different patterns of defection

They use what they call normalized generosity in order to compute the probability of a peer cooperating with another peer. Generosity is the measuring of the benefit an entity has provided relative to the benefit it has consumed, given by Formula 1:

$$g(i) = {}^{p_i}/_{c_i}$$

Formula 1. Generosity formula.

Then, the normalized generosity measures one peer's generosity relative to another peer's generosity as shown in Formula 2:

$$g_j(i) = {}^{g(i)}/_{g(j)}$$

Formula 2. Normalized generosity formula.

Using these concepts, the authors show through a game theoretic approach to cooperation in peer-to-peer networks how their approach addresses the challenges imposed by P2P systems, including large populations, high turnover, asymmetry of interest and zero-cost identities. Their results prove that the adoption of shared history and discriminating server selection techniques can mitigate those aforementioned challenges and also that cooperation can be established even in the presence of zero-cost identities through the use of an adaptive policy towards strangers. Finally, colluders and traitors can be kept in check via subjective reputations and short-term history respectively.

The work by Koutrouli et al. [11] deals with the attacks and misbehaviours suffered by most P2P systems nowadays. Free riding and badmouthing are two of the most important problems affecting P2P systems and the authors argue that providing incentives can help reducing those problems. A credit-based recommendation exchange is proposed in order to provide incentives for honest participation in P2P reputation systems where payments for recommendations are based on the trustworthiness of peers regarding the accuracy of the recommendations they give. The payment value (v) is a virtual amount, which will be transferred between peers' virtual accounts but without performing a real currency transfer, and the formula is designed so the recommendation reputation of the buyer and the seller impact the calculation. If the former is higher than the latter, the payment will be always lower than one, while in the opposite situation it will be always higher than one. In their system, every peer starts with an initial account balance, which determines the highest amount of recommendation exchange transactions that a new peer can get involved before running out of credit. The account balance of a peer is updated after each exchange and the maximum amount of times that a peer which has always worst recommendation reputation value than its peers can participate in the system is limited due to the nature of the calculation formula. Finally, they also implement a recommendation exchange protocol using an overlay that implements the payments. Their simulation results show that the dishonest recommendation behaviour results in non-participation in the reputation system, whereas honest recommendation behaviour results in the maximum utility of the reputation system, thus effectively providing an incentive for honest recommendations and good behaviour.

Finally in Aldini et al. [12], the authors propose that the success of user-centric networks strongly depends on the willingness of the participants to cooperate and that incentives can help in encouraging users to cooperate. To this end, reputation-based

incentives and remuneration incentives are introduced to increase the users' motivation and to discourage selfish behaviours. In their work, quantitative properties of cooperation incentives are defined and analysed through model checking. Their model considers users providing services, which are called providers and users receiving services, which are called requesters, presenting four phases of cooperation:

1. discovery and request
2. negotiation
3. transaction
4. evaluation and feedback

Their reputation system defines cooperative attitude, which depends on dispositional trust and on service trust level, which represents the threshold under which the service is not accessible. For the service request to be accepted by a given node, the trust computed for the provider should be higher than the service trust level threshold. The authors also introduce a virtual currency system where reputation-based and reward-based incentives are combined by including the trust level T of the provider towards the requester as a parameter affecting the cost of the negotiated service. Cost is calculated as shown in Formula 3:

$$C(T) = \begin{cases} C_{min} + \dfrac{C_{max} - C_{min}}{T'} \times \left(T' - T\right) \; if \; T < T' \\ C_{min} \; if \; T \geq T' \end{cases}$$

Formula 3. Cost computation formula.

Where the parameters are: C_{min}, which is the minimum reward asked by the provider regardless of his/her trust on the requester, C_{max}, which is the maximum reward asked to serve untrusted users, and T', which is the trust threshold above which the minimum cost is applied to the requester. Finally, they prove through Markov decision process analysis that mixing incentive strategies such as reputation and reward proves effective in inducing cooperative behaviours and also that cooperation incentives favour both requesters and providers, as honest requesters get services at a lower price and reputation and cooperative behaviours impact earnings in providers.

Decentralized sustainable wireless networks

In Hubaux et al. [13], the authors tackle the problematic of security in mobile ad-hoc networks, as it is normally quite difficult to achieve given the vulnerability of the links, the limited resources available and the dynamically changing topology among others. In their work they start by defining the threats that affect the most these networks and which can be directed not only against the basic mechanisms but also against the security mechanisms themselves. Regarding the vulnerabilities affecting basic mechanisms, they highlight the risk of nodes being hijacked, eavesdropping, active interferences as the communication are carried out over the air, non-cooperative nodes, vulnerabilities related to the routing mechanisms and malicious neighbour discovery. Regarding the vulnerabilities affecting the security mechanisms directly, they address mainly the risks of cryptographic keys being compromised or replaced with other keys. In order to protect the basic mechanisms their choice is to use tamper resistant

hardware and smart cards to protect the cryptographic information, while also aiming to protect the routing mechanisms by using watchdogs and rating paths in combination with intrusion detection systems (IDSs). Finally, in order to enforce the service, they use a virtual currency called *nuglets* as a cooperation incentive.

The work in Pirzada et al. [14] states that the execution and survival of an ad-hoc network is exclusively dependent on the cooperative nature and trustworthiness of its nodes. The problem they find is that it is actually this same dependency on intermediate nodes what makes an ad-hoc network vulnerable to passive and active attacks carried out by malicious nodes. There are a good amount of protocols that have been developed to secure ad-hoc networks using cryptographic schemes, but almost all of them rely on the presence of a central trusted authority and as the authors state, dependence on a central trust authority is an impractical requirement for ad-hoc networks as their dynamic topology and spontaneous nature makes this highly unfeasible. In order to tackle this problem, the authors propose a model which implements trust-based communication in ad-hoc networks and that also proves that a central trusted authority is not always a strong requirement. Their model introduces the notion of belief and provides a dynamic measuring of reliability and trustworthiness in a given ad-hoc network. Their trust model uses an adaptation from Marsh's [15] work, but modified in order to be used in ad-hoc networks. In their work, the authors make use of trust agents that reside on each of the network nodes and each agent operates independently and maintains its individual perspective of the trust hierarchy. In the regular operation cycle of an agent, it first gathers data from events in all the states, then it filters it and assigns weights to each event and finally it computes different trust levels based upon them. They after use this trust model to enhance the dynamic source routing (DSR) protocol in order to find the trustworthiest routes from one node to another, improving the survivability of the network by avoiding routes containing malicious nodes. Also, as the model presented operates passively and has minimal computation and energy requirements, it also improves the sustainability of the network by saving the energy and bandwidth of the nodes.

To finalize, Xing et al. [16] focus on the analysis of network survivability in the presence of misbehaving nodes and failures. In order to tackle this problem they propose a novel semi-Markov process model to study the evolution of nodes' behaviours and as an immediate application of the proposed model they investigate the problem of node isolation where the effects of Denial-of-Service (DoS) attacks are considered. The authors also find that the network survivability degradation is directly proportional to the increase of misbehaving nodes and that moreover DoS attacks have a significant impact on the network survivability, especially in dense networks. To finalize, they validate their proposed model and their analytical results using numerical analysis and showing the effects of node misbehaviours on both topological survivability and network performance.

Identifying the research gap

Table 1 presents the comparison of all the frameworks and metrics analysed in the previous sections regarding three main features.

As can be seen from the previous table comparing and summarizing the main characteristics of the metrics and frameworks that have been analysed, none of them comply

Table 1 Frameworks and metrics comparison

Framework/metric	Fully decentralized	Sybil resistant	Incentives
Certain trust	✗	✗	✗
Martucci et al.	✗	✓	✗
Appleseed	✓	✗	✗
Eigen trust	✓	✗	✓
Feldman et al.	✓	✗	✓
Credible recommendations	✓	✗	✓
Bogliolo et al.	✓	✗	✓
Nuglets	✗	✓/✗[a]	✓
Pirzada et al.	✓	✗	✓
Xing et al.	✓	✗	✗

[a]While it provides means to greatly reduce Sybil attacks close to a full extent, it is not 100 % Sybil resistant (i.e. there are mechanisms to try to deter Sybil nodes or to detect them, but those not deterred or detected still are able to cheat)

at the same time with the three basic characteristics that we are aiming for in our framework.

While the literature reviewed presents frameworks which always comply with one or more of the desirable characteristics that we deem as desirable for such decentralized user-centric network environments, none of them is truly resistant to Sybil attacks, and some frameworks or metrics, even when applied to decentralized wireless environments, still rely in some centralized elements or centralized bootstrapping steps. Moreover, to the best of our knowledge, there is no or little work done to study how trust management can be coupled with cooperation incentives in order to empower the latest, in a fully decentralized way and being fully Sybil attack resistant and this is the gap we are aiming to fill with our own framework, which is presented in the following section.

Framework model and design

After having identified and precisely defined the gap, we are now going to present and thoroughly describe our framework, which covers this previously identified gap by bringing together an attack resistant trust metric and a cooperation incentives schema in the form of points as a form of reward in a fully decentralized and user-centric fashion. Our framework has three main core component or building blocks as can be seen in Fig. 1.

The trust manager is in charge of computing trust levels for the different nodes that the user has to interact with, the cooperation manager is dedicated to provide and compute incentives and rewards and finally the identity manager is the responsible entity for generation virtual identities in the form of crypto-ids for each of the nodes or users in the system.

In the following sections we will first introduce the formal definitions tied to this framework and afterwards we will describe each of the main building blocks and their subcomponents in detail as well as their functionality.

Formal definitions

This section introduces the formal definitions for the components of our framework.

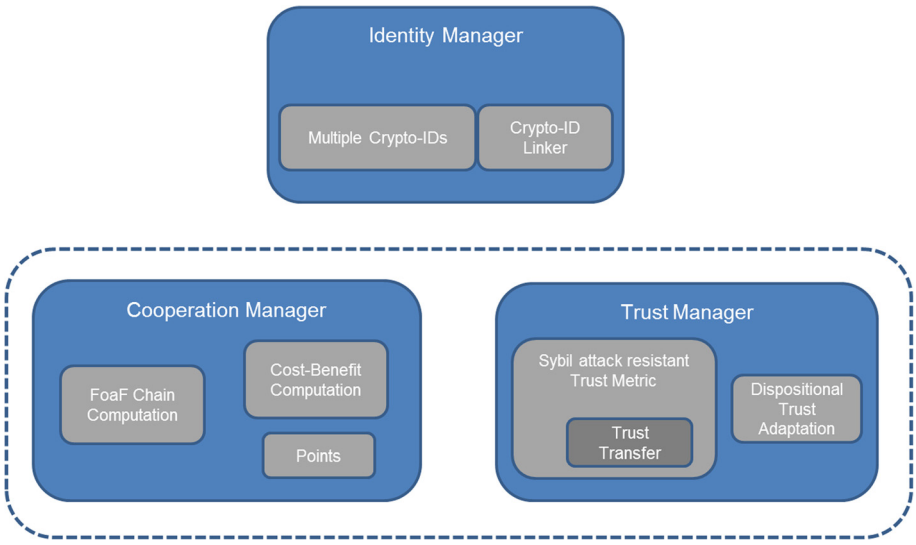

Fig. 1 The framework

User

We define a user on our framework as any entity, be it human or not, which controls and is able to use one or more devices or nodes to interact with the system.

Users can be of two kinds, honest or malicious, which are defined like follows:

- An *honest user* is a user that interacts with the system through its device(s) or node(s) in a rightful way, behaving properly, not taking any advantage from whatever flaw the system might have and always providing a service with equal or higher quality than agreed with the counter party.
- A *malicious user* on the other hand, is the user which when interacting with the system through its device(s) or node(s) tries to take unfair advantage from the system, exploits it and/or provides services with lower quality than agreed or doesn't provide the service at all.

A user is represented by one or more virtual identities in the form of crypto-ids.

Device or node

We define a device or node in our framework as any element that is owned by a user, as defined previously, and that enables a user to interact with the system. The term device and node are interchangeable in our framework and can be used indistinctly.

A device or node can be seen as honest or malicious depending on how it interacts with the system, but in reality it is the user owning and controlling it who is malicious or not, as the devices only act as the mean to interact with the system for a given user, being the latter the responsible for the good or the bad use of them. A node can be of two types:

- **Requester:** a node requesting a service.
- **Provider:** a node providing a service.

Manager

We define a *manager* as a core element to our framework which has a well-defined purpose and a set of basic operations and functionalities. A manager can be of three types, which will be defined and explained later in the next sections, *trust manager*, *co-operation manager* or *identity manager*.

Each of the managers in the framework is responsible for the part that gives its name and all of them are necessary for the well-functioning of the framework.

Attack

We define an attack in our framework as any intent of directly or indirectly exploiting a vulnerability of the system to gain an unfair advantage over it.

Trust manager

The first main building block in our framework is the trust manager. The trust manager is in charge of managing the dispositional trust adaptation for the user's device(s) and for providing trust computation in order to assign trust levels to other nodes in the system using a Sybil resistant trust metric as can be seen in Fig. 2.

Following, we will explain in detail each of the components of the trust manager.

Dispositional trust adaptation

User-centric networks are supported both by static, fully dedicated nodes as well as by nodes provided by end-users on the fly. Since some nodes are carried by Internet end-users, their networking composition, surrounding environment and organization can rapidly change.

Dispositional trust reflects the disposition of a certain individual to trust or not "*per-se*", and it is mostly represented by a fixed value which doesn't change over the time. Contrary to this, we think that dispositional trust is a value that, as far as it is modelling a part of human trust, it is also subject to changes. When trying to copy human behaviour and translating it into computational trust notions and representations, we think that it is important to take into account, that the disposition to trust usually changes over the time according to the situation and surrounding environment and the interactions with it. As such, a hostile environment might turn an open-to-trust individual into a distrustful one, and on the other hand, a reliable environment can turn a

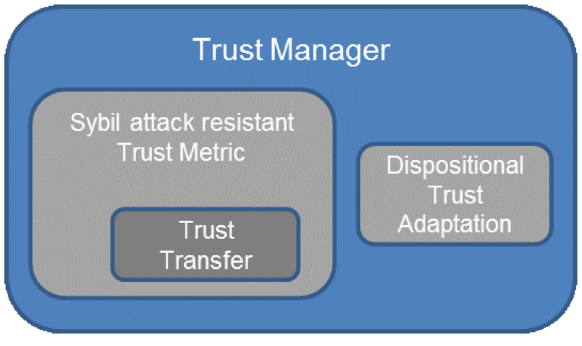

Fig. 2 Trust manager components

distrustful individual into a more willing-to-trust one. The aim of adapting dispositional trust is twofold:

- Firstly, it can be helpful when trying to protect a node in such dynamic and highly changing environments, where often nodes have not had yet an interaction with many other nodes in the system and thus the uncertainty is high.
- Secondly it can help the metric to converge faster to realistic trust values in extreme environments, i.e. environments with a big majority of misbehaving nodes.

We define dispositional trust $D_T \in [-1, 1]$ where:

$$-1 = \textit{full distrust and}$$
$$1 = \text{full trust}$$

Dispositional trust adaptation is based on the adaptation rate and adaptation step intervals defined following. Adaptation rate defines how many malicious or honest interactions a node needs to have in order to re-adapt its dispositional trust and adaptation step defines the numeric amount in which the dispositional trust level is adapted, i.e. how much it decreases or increases after a certain amount of malicious or good interactions with other nodes in the system. We define adaptation rate A_R and adaptation step A_S as:

$$A_R \in \{1, 2, 3, 4\} \textit{ and } A_S \in \{0, 0.2, 0.3, 0.4\}$$

We normalize D_T as D_{TN} into [0, 1], when a value in between 0 and 1 rather than in between −1 and 1 is needed, according to Formula 4:

$$D_{TN} = 0.5 + \frac{D_T}{2}$$

Formula 4. Dispositional trust normalization.

Finally, we adapt dispositional trust as shown in [17]; every time a requester accumulates a number of malicious interactions M_I or good interactions G_I then:

$$D_T \begin{cases} \textit{if } M_I \geq A_R & \textit{then } D_T = D_T - A_S \\ \textit{if } G_I \geq A_R & \textit{then } D_T = D_T + A_S \end{cases}$$

Formula 5. Dispositional trust adaptation.

The results of adapting dispositional trust and the values chosen for the parameters were studied, simulated and evaluated in a previous paper by the authors [17].

Trust transfer

Trust transfer [3] has been proven to protect against Sybil attacks when pieces of evidence are limited to direct observations and recommendations based on the count of event positive outcomes. Trust transfer implies that recommendations move some of the trustworthiness of the recommending entity to the trustworthiness of the trustee. This approach is particularly efficient for our system, as besides assessing trust we can use the metric to reward in the form of trust points the agents that share their Wi-Fi

connectivity, effectively combining trust management with cooperation incentives as will be explained in following sections.

Based on

Figure 3, Trust Transfer works in the following manner:

1. The subject (S) requests an action, requiring a certain amount of positive event outcomes - trustworthiness is based on event outcomes count in Trust Transfer – in order for the request to be accepted by the trustor (T).
2. If S has not enough trust, T queries its contacts to find recommenders (R) willing to transfer some of their positive event outcomes count to S.
3. If the one or more contacts have interacted previously with S and the contacts' trust balance with T allows it to permit to transfer an amount of the recommender's trustworthiness in S, the contact agrees to recommend the subject. It queries the S on whether it agrees to lose that same amount of trust on the recommender's side.
4. Subsequently S returns a signed statement, indicating whether it agrees or not.
5. Finally, R sends back a signed recommendation to T, indicating the trust value it is prepared to transfer on behalf of S, including the signed agreement of S.

We define trust level from $A \rightarrow B$ as T_{AB}, direct trust from $A \rightarrow B$ (direct observations) as TD_{AB}, recommendations as TR and trust threshold T_T, all $\in [0, 1]$.

Cooperation manager

The second main building block in our framework is the cooperation manager. The cooperation manager is in charge of computing the cost and benefit balance of a given action or service exchange, managing the node's points balance and searching and computing friend-of-a-friend chains in order to find potential point lenders as can be seen in Fig. 4.

Following, we will explain in detail each of the components of the cooperation manager.

Cooperation incentives: points and other rewards

In order to foster interaction amongst users in a collaborative environment such as the one described in this article, there is a need to offer incentives to the users besides providing them with the appropriate safety features such as a robust trust metric.

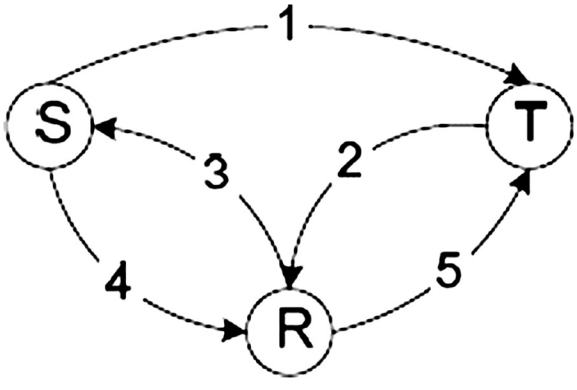

Fig. 3 Trust transfer process [2]

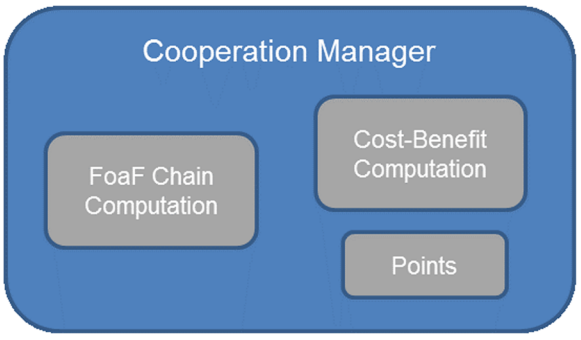

Fig. 4 Cooperation manager components

Trust Transfer can effectively be used as a cooperation incentive enabler, by using its trust points as the de facto "currency" in order to be able to use the services other users have to offer, in this case Wi-Fi connectivity sharing. By awarding trust points to the service provider proportionally to the duration of the Wi-Fi sharing period, we foster cooperation among users as not only the trust points reflect the good behaviour of the user giving her a good reputation, but also enable her to in turn obtain Wi-Fi connectivity when roaming or being out of data by using those trust points earned previously in order to pay for the service. The more you share in the system, and the more different users you share with, the easiest will be to in turn find another user which will accept your trust points as payment, be it because of having interacted directly with her or using trust transfer mechanisms to find another user who can lend the service requester those needed points as explained in the previous section. We reckon that these incentives are limited by your own circle of direct interactions and acquaintances inside the system, and this is why we exploit another capability of trust transfer, which is being able to transfer trust points through chains of trust with multiple hops, as explained in the next section.

We define cooperation points as the points given, in the case of a service requester, or gained, in the case of a service provider, when a service exchange happens in between two nodes.

Points are defined in units in the range of $[1, +\infty[$, and being 1 the minimum amount of points able to be gained or paid for a given service.

Modelling cooperative behaviours

For a cooperation incentives schema to work, a basic premise that needs to hold is that the potential benefits obtained from behaving good is greater than the potential costs incurred when performing an action in the system. In our collaborative sharing service, we model our cooperation incentives schema in form of credits, which can be obtained when sharing a Wi-Fi access and spent when using other user's Wi-Fi access. Then, we compute the profit of cooperating according to Formula 6:

$$P_{(rofit)} = \frac{B_p}{C_p}$$

Formula 6. Profit of cooperation.

In this equation, B_p stands for the potential benefit a user can obtain when behaving good and C_p stands for the potential cost of performing a certain action, where P is

contained in the interval *[0, +∞[*. We then take the decision on whether to cooperate or not according to Formula 7:

$$D_{(ecision)} \begin{cases} if\ P > 1,\ \ cooperate \\ else,\ \ non-cooperative\ behaviour \end{cases}$$

Formula 7. Decision to cooperate.

Small world networks

To empower the cooperation incentives provided by Trust Transfer and the trust points, some other mechanism in order to extend the usefulness of those points needs to be introduced, as Trust Transfer contemplates mainly that trust points are to be used "one-to-one", or as most with one degree of indirection. This means that in a scenario where several strangers are supposed to cooperate and to share services, it would be difficult to spend those points as the likeliness of finding in the same environment another user which one has already interacted with, or as most within one degree of separation is highly unlikely.

In order to overcome this limitation, we have explored the probabilities of finding longer "friend-to-friend" chains, applying the principles of small worlds [18] and degrees of separation. For the sake of simplicity, we assume that most of the system's users come from networks which are already highly connected, such as Facebook.

Social networks like Facebook have been proven to have a degree of separation of around 4.76 to 6 with almost a 100 % of probabilities [19, 20]. The problem of finding the probabilities for a subset of a small world network to find a chain of 6° of separation or less can be modelled as random node failures (different from targeted attacks) in the complete network until we are left with the desired amount of nodes, which would be our subset of the small word network. In order to model a social network like Facebook, we need to use a scale-free network that exhibits both short paths and high clustering degree. Such a network can be modelled by using a Klemm and Eguíluz (KE) [21] network, which is a type of scale-free network which complies with both properties.

We define degrees of separation in our framework as it is defined in small world networks' mathematical models [22], being it the distance in between two given nodes in the user-centric network environment of our system. The formal definition of distance (***d***) can be seen in Formula 8:

$$d = 1 + \frac{\ln N}{(\ln k + (\ln k - 1))}$$

Formula 8. Distance between two nodes.

Where ***N*** is the number of nodes conforming the user-centric network, and ***k*** is the number of links per node to other nodes in the network.

Friend-of-a-friend chains

While the most used metrics to determine the properties of a network are L (characteristic path length) and C (clustering), those can produce misleading results when used to re-evaluate such properties when eliminating large portions of random nodes, as disconnected or isolated users or small unreachable clusters can skew the results. It is thus a better estimate of the properties of a network, as stated in Crucitti et al. [23], the one

produced by the global and local efficiency (E_{glob} and E_{loc}). The efficiency of a network is defined as the effectiveness of the network to propagate information both globally and locally, meaning the possibility of finding a path in between two nodes of that network for the information to propagate. Those definitions can be modelled mathematically as seen in Formula 9 and Formula 10.

$$E_{glob}(G) = \frac{1}{N(N-1)} \sum_{i \neq j \in G} \frac{1}{d_{ij}},$$

where d_{ij} is the shortest path in between nodes i and j.

Formula 9. Global efficiency formula.

$$E_{loc} = \frac{1}{N} \sum_{i \in G} E(G_i)$$

Formula 10. Local efficiency formula.

Taking Formula 10 into account, and applied over a network inducing random failures and targeted attacks, the authors in [23] have come up with the results that can be seen in Fig. 5.

As we can see in the previous graphs, until the network is not at least a 20 % of the original size, the efficiency or clustering size is not big enough to even consider it a functioning network. Nevertheless, there are other aspects that have not been taken into account in the purely mathematical demonstration:

- Facebook is especially high clustered (much more than any of the networks in the previous results), to which one could argue that the removal would not impair the network as badly as that.
- When users decide to adopt a system which is collaborative and based in friendships, most likely it will be adopted in an «epidemic» way, on which friends and friends of friends would install it, leading to an also highly clustered and connected sub-network.
- The interactions between disconnected users while using our system, would in the long run create a small world by itself.

In our simulations, we apply these same principles and we calculate for a given user base population, how quick the full system would bootstrap and which is the minimum amount for such a user base which would enable reasonable probabilities of finding such FOAF chains so the cooperation incentives are more useful and in turn, encourage the users to cooperate and behave properly.

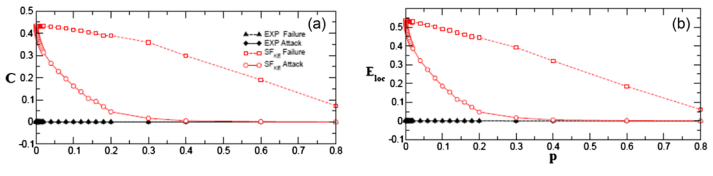

Fig. 5 Clustering and efficiency loss for percentages of random failure in nodes and targeted attacks [23]

Identity manager

The third main building block in our framework is the identity manager. The identity manager is in charge of creating virtual identities for the user and managing them in order to be able to link trust levels to identities and to combine if needed identities in order to be able to exhibit a higher trust level by linking those virtual identities as can be seen in [23] Fig. 6.

By design choice, we are going to use in our framework the concept of *crypto-id* as virtual identity as introduced in Seigneur's PhD thesis [2]. Simply put, a crypto-id is a piece of cryptographic material belonging to a user, which is then hashed to produce a unique identifier that can represent that user. As cryptographic piece we are going to use the public key of a public-private key pair, which can be generated at will by each identity manager belonging to each of the users in the system, and then hash that public key in order to produce the crypto-id itself.

In our framework, a crypto-id is defined according to the following general Formula 11:

$$CryptoID_A = f_{HASH}\left(PubKey_A\right)$$

Formula 11. Crypto-id creation.

Where $f_{HASH}()$ represents any available hashing function or algorithm such as SHA-1/2/3, MD5, etc., and $PubKey_A$ is the public key from the private-public key pair from node or device A.

Following, we will explain in detail each of the components of the identity manager.

Multiple crypto-ids

Given that strong enrolment in a centralized authentication manner is not available in our framework, as we want it to be fully decentralized, we cannot rely on each of the users to have a validated and unique identity. On the other hand, the usage of multiple crypto-ids facilitates attacks at the identity level on trust management, for example, as said before voting several times with different virtual identities owned by the same user, namely Sybil attacks. In order to solve this issue, rather than trying to forbid the users to create multiple virtual identities, our framework allows them to create many virtual identities based on crypto-ids and mitigates potential attacks by providing an attack-resistant trust metric as introduced in previous sections, Trust Transfer. Although this approach is more difficult to achieve than unique crypto-ids at the trust management level, it allows for the creation of fully a decentralized user-centric network

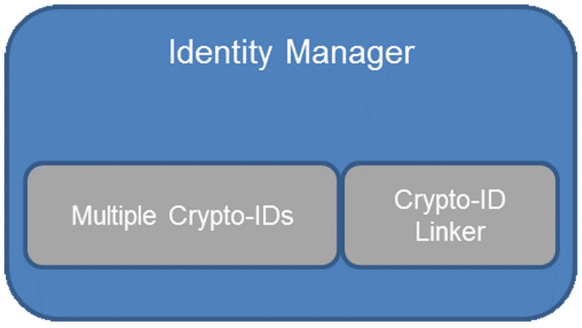

Fig. 6 Identity manager components

environment framework, and also the improvement of privacy protection by default, as the users can choose to split their actions among different pseudonyms or crypto-ids, making it more difficult to have a complete view of the actions executed by one user and find her real-world identity through action linking or extensive data collection or mining.

To create multiple crypto-ids, the user can generate multiple key-pairs that will correspond to different pseudonyms that she can use in different situations during her interaction with the decentralized environment. Those key pairs would be used to sign requests and messages and to identify herself.

Crypto-id linking

Still according to Seigneur's PhD thesis [3], in order to implement a mechanism balancing trust with privacy, as we said in the previous section we allow users to freely create pseudonyms identified by the crypto-id, i.e., the hash of the public key of a locally generated asymmetric cryptography key pair. Then, depending on the context, one or another pseudonym can be used to carry out actions logged as events signed with the private key of the pseudonym.

If needed, one or several pseudonyms could also be linked together in order to increase the number of known actions and potentially increase the trust in the linked entity assuming that all these actions had a positive outcome. As each crypto-id is able to sign, two crypto-ids can both sign a special message, called "crypto-ids linking message", saying that they are linked together meaning that they have the same owner. In this way, the trust level linked to each of the crypto-ids is proven to belong to the very same user, allowing her to carry out an action that perhaps couldn't be taken with the trust level linked to one of her single crypto-ids alone.

Methods

The model has been simulated using AnyLogic [24]. AnyLogic is a simulation tool that supports System Dynamics, Process-centric (Discrete Event), and Agent Based modelling, based on the Eclipse platform. The flexibility of its modelling language provides the opportunity to capture the complexity and heterogeneity of a given system to any desired level of detail, and its object-oriented model design paradigm provides for modular, hierarchical, and incremental construction of large models. The simulation environment corresponds to a real world area, which is the airport of the city of

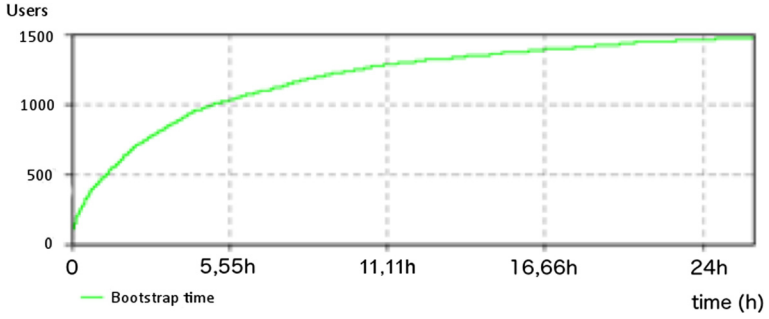

Fig. 7 Bootstrap time with 200,000,000 system users worldwide

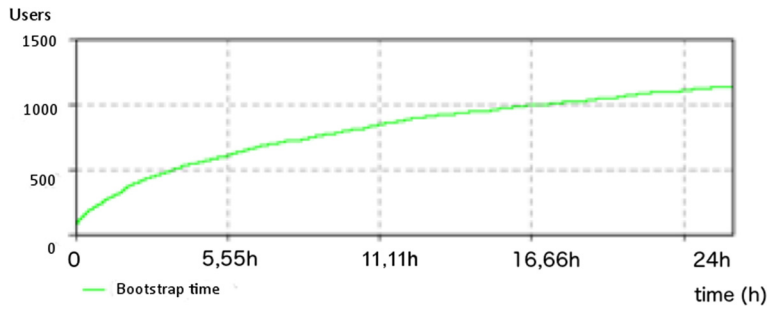

Fig. 8 Bootstrap time with 20,000,000 system users worldwide

Geneva, Switzerland. The environment has been modelled respecting the real dimensions of the airport, and also the real proportions of both local and foreign travellers and permanent workforce of the airport [25]. The exact details of the simulation are as follows:

- 450 m long and 150 m wide, spanning 3 floors of this same size
- Around 13 million passengers in 2012, from which 55 % are foreigners and 45 % are locals.
- 840 staff and permanent workers (working in shifts).

Taking into account this previous data, each of the simulation runs has been done with 3000 agents that simulate passengers (both local and foreign in the proportions previously mentioned) and 280 workers (assumed always locals) at any time, included in those numbers. To make the scenario as realistic as possible, agent renewal happens with a normal distribution with an average of 2 h in order to simulate the passengers leaving and new ones arriving. Workers are also renewed in 8 h shifts. We assume that locals have an average of 15–20 friends (acquaintances or previously interacted users) and foreigners an average of 2. All local workers are known to each other.

Results and discussion

In this section, we proceed to present the details of the simulation environment, and the results obtained from running those simulations, both in terms of bootstrapping time and user data depletion times.

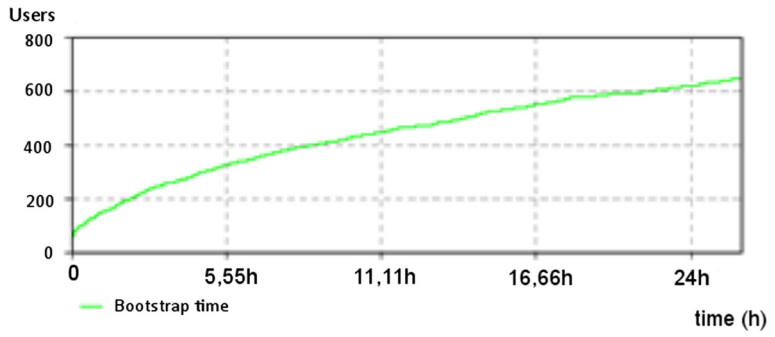

Fig. 9 Bootstrap time with 2,000,000 system users worldwide

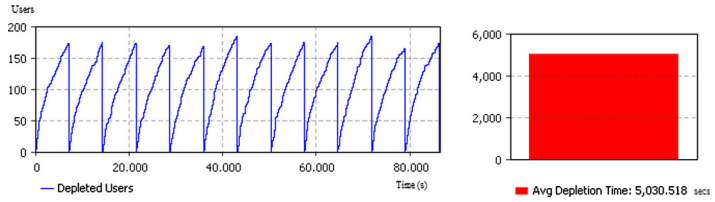

Fig. 10 Amount of depleted users and average depletion time with 200,000,000 system users worldwide

Simulation results

In order to study the feasibility of the system, we have run several simulations each with a different user base for the system. This user base is a key point, as it will determine the threshold from which the system might be usable both from the bootstrapping point of view and from incentives perspective. Note that when we talk about user base (or system users), we are not talking about the amount of agents in the simulation, which are fixed according to the criteria mentioned in the previous section, but to the total amount of users in the world using this system. This user base is what enables the probabilities of finding long FOAF chains in order to enhance the cooperation incentives provided by Trust Transfer. Each simulation runs for a real-world whole day, measured in seconds (86400 s).

Bootstrapping measurements

For the system to be usable, the bootstrapping time needs to be as low as possible in order for the foreigner passengers to be able to connect to locals while in their short time at the airport. We consider that the system is bootstrapped if half of the agents that can provide connectivity have successfully shared at least once their Wi-Fi with a foreign or a local agent that might have run out of data. For each of the graphs presented below, the Y axis represents amount of agents and the X axis simulation time, measured in seconds. We have run the simulation for different sizes of user base population, ranging from 2 million system users to 200 million system users with an intermediate simulation accounting for a 20 million system user base. The results can be seen in Figs. 7, 8 and 9.

As can be seen from the results, if we want to achieve the aforementioned objective of half the agents having shared their Wi-Fi with foreigners in a reasonable time, the only configuration achieving this is the one with 200 million system users. This accounts for 750 agents in roughly 7,500 to 8,000 s, which is close to the average time for agent renewal in the simulation, making it a feasible time for the system to be bootstrapped.

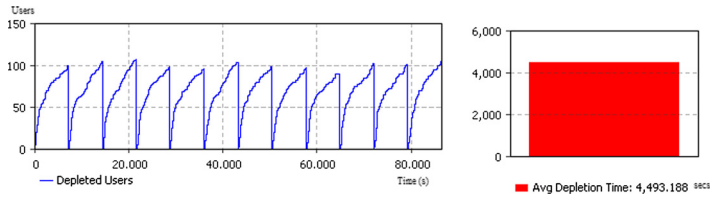

Fig. 11 Amount of depleted users and average depletion time with 20,000,000 system users worldwide

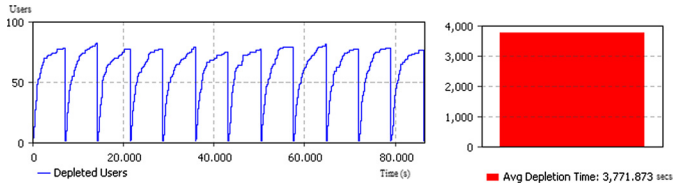

Fig. 12 Amount of depleted users and average depletion time with 2,000,000 system users worldwide

Resource depletion measurements

Another interesting measurement for us is how quick users run out of data capacity, and which is the average time that it takes for a given user to be depleted of her data capacity.

We have run the simulation for different sizes of user base population, ranging from 2 million system users to 200 million system users with an intermediate simulation accounting for a 20 million system user base. For each of the figures, the left-hand graph represents the amount of data depleted users in a given point of time, being the Y axis the amount of users and the X axis the time in seconds, and the right-hand graph represents the average time that took for those users to be depleted of their data capacity, measured in seconds. The results can be seen in Figs. 10, 11 and 12.

As can be seen from the results, with a smaller system user base the amount of depleted users in each renewal period is also smaller, but the average depletion time for each of those agents is lower as well. The implications of this will be discussed in the next subsection.

Discussion

From the previous simulation runs carried out, we can summarize the results in Table 2.

As can be seen from the summary, the bigger the system user base is, the better the results, both in terms of bootstrapping time and depletion measurements.

Regarding bootstrapping results, there is a critical user base needed in order to find a chain connecting a service requester with a service provider. As can be seen in Table 2, 200 million users worldwide fulfils this critical mass needed, accounting for the shorter bootstrapping time, as it is more likely to find users who can transfer some trust points from one end to the other and thus enabling cooperation in between the two users. It is also worth to note that with the use of the system the probabilities of finding users from which to get points increases as the interactions in between agents increase. This translates into an increase of the probabilities of finding a chain of agents from which to get points lent from one end to the other by 0.1 % per interaction per agent. Arguably, it could be said that a 20 million user base could be enough to obtain a reasonable

Table 2 Summary of results

User base (in millions)	Bootstrap time (in hours)	Depleted users per renewal period	Average depletion time (in hours)
200	2.26	175–185	1.39
20	7.87	95–105	1.24
2	>24	75–85	1.04

bootstrapping time (~7.8 h), but with a user base closer to 200 million we can achieve times which are closer to the agent renewal time in our scenario, making it closer to the ideal situation.

Regarding data depletion, as true as it is that with smaller system user amounts there are less agents that get depleted from their daily quota allowance, this is due to the fact that also there are less agents being able to connect and to allow connections in order to share Wi-Fi as it is more difficult to find a longer user chain to transfer trust points. On the other hand, it can also be seen that the average time taken to deplete a user from her daily data quota is higher the bigger the user base is, meaning that even though more users are depleted in each agent renewal period, those users take longer to be depleted due to the higher amount of agents being able to share their Wi-Fi connection. It is also worth to note that even being a higher number of depleted users, those account only for ~10 % approximately of the total amount of agents being able to share their Wi-Fi connectivity (175–185 out of 1500).

Conclusions

In this paper, we have proposed extending trust management with cooperation incentives for collaborative Wi-Fi sharing and we have identified the most important shortcomings affecting these kinds of frameworks. Through the use of trust management and cooperation incentives we have put in place measures to eradicate or mitigate all of them, and finally, we have shown through simulation the effectiveness of the combination of our trust and cooperation incentives schema in regards of bootstrapping time and data depletion, linking it to the amount of users the system has and finding which is that ideal amount.

It is left for future work to compare our trust metric and incentives schema with other trust metrics such as EigenTrust or Appleseed.

Competing interests

The authors declare that they have no competing interests.

Authors' contribution

CBL designed the framework model and carried out the simulation and evaluation. J-MS contributed in the state of the art and the framework design and helped drafting the manuscript. Both authors read and approved the final manuscript.

Received: 18 December 2014 Accepted: 22 June 2015
Published online: 14 August 2015

References

1. The World in 2014: ICT Facts and Figures, ITU. [Online]. Available: http://www.itu.int/en/ITU-D/Statistics/Pages/facts/default.aspx. [Accessed: 27-May-2014]
2. Douceur J (2002) The sybil attack. In: Peer-to-Peer Systems. Springer, Heidelberg, pp 251–260
3. Seigneur JM (2005) Trust, security and privacy in global computing. In: PhD Thesis. Trinity College, Dublin
4. W3C RDF Friend of a Friend (FOAF) vocabulary. The Friend of a Friend (FOAF) RDF vocabulary, described using W3C RDF Schema and the Web Ontology Language
5. Ries S (2007) Certain Trust: A Trust Model for Users and Agents. In: Proceedings of the 2007 ACM Symposium on Applied Computing, New York, NY, USA., pp 1599–1604
6. Martucci LA, Ries S, Mühlhäuser M (2011) Sybil-free pseudonyms, privacy and trust: identity management in the internet of services. J Inf Process 19:317–331
7. Ziegler CN, Lausen G (2004) Spreading activation models for trust propagation. In: e-Technology, e-Commerce and e-Service, 2004 IEEE International Conference on., pp 83–97
8. Levien R (2009) Attack-resistant trust metrics. In: Computing with Social Trust. Springer, Heidelberg, pp 121–132
9. Kamvar SD, Schlosser MT, Garcia-Molina H (2003) The Eigentrust Algorithm for Reputation Management in P2P Networks. In: Proceedings of the 12th International Conference on World Wide Web, New York, NY, USA., pp 640–651
10. Feldman M, Lai K, Stoica I, Chuang J (2004) Robust Incentive Techniques for Peer-to-peer Networks. In: Proceedings of the 5th ACM Conference on Electronic Commerce, New York, NY, USA., pp 102–111

11. Koutrouli E, Tsalgatidou A (2013) Credible Recommendation Exchange Mechanism for P2P Reputation Systems. In: Proceedings of the 28th Annual ACM Symposium on Applied Computing, New York, NY, USA., pp 1943–1948
12. Aldini A, Bogliolo A (2012) Model Checking of Trust-Based User-Centric Cooperative Networks, presented at the AFIN 2012. The Fourth International Conference on Advances in Future Internet, pp. 32–41
13. Hubaux J-P, Buttyán L, Capkun S (2001) The Quest for Security in Mobile Ad Hoc Networks. In: Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking & Computing, New York, NY, USA., pp 146–155
14. Pirzada AA, McDonald C (2004) Establishing Trust in Pure Ad-hoc Networks. In: Proceedings of the 27th Australasian Conference on Computer Science - Volume 26, Darlinghurst, Australia, Australia., pp 47–54
15. Marsh SP (1994) Formalising Trust as a Computational Concept, Dissertation. Department of Mathematics and Computer Science, University of Stirling, Stirling
16. Xing F, Wang W (2010) On the Survivability of Wireless Ad Hoc Networks with Node Misbehaviors and Failures. IEEE Trans Dependable Secure Comput 7(3):284–299
17. Lafuente CB, Seigneur J-M (2013) Dispositional Trust Adaptation in User-Centric Networks. In: Advanced Information Networking and Applications (AINA), 2013 IEEE 27th International Conference on., pp 1121–1128
18. Milgram S (1967) The small world problem. Psychol Today 2(1):60–67
19. Backstrom L, Boldi P, Rosa M, Ugander J, Vigna S (2012) Four Degrees of Separation. In: Proceedings of the 4th Annual ACM Web Science Conference, New York, NY, USA., pp 33–42
20. Ugander J, Karrer B, Backstrom L, Marlow C (2011) The anatomy of the facebook social graph. ArXiv Prepr. ArXiv11114503
21. Klemm K, Eguíluz VM (2002) Highly clustered scale-free networks. Phys Rev E 65(3):036123
22. Watts DJ, Strogatz SH (1998) Collective dynamics of 'small-world' networks. Nature 393(6684):440–442
23. Crucitti P, Latora V, Marchiori M, Rapisarda A (2003) Efficiency of scale-free networks: error and attack tolerance. Phys Stat Mech Its Appl 320:622–642
24. Multimethod Simulation Software and Solutions. [Online]. Available: http://www.anylogic.com/. [Accessed: 27-May-2014]
25. Genève Aéroport - Statistics. [Online]. Available: http://gva.ch/en/desktopdefault.aspx/tabid-244/. [Accessed: 27-May-2014]