Using Trust and Local Reputation for Group Formation in the Cloud of Things

G. Fortino[a], F. Messina[b], D. Rosaci[c], G.M.L. Sarné[d]

[a]*Department DIMES, University of Calabria, Via P. Bucci, cubo 41c, 87036 Rende (CS)*
[b]*Department DMI, University of Catania, Viale Andrea Doria 6, 95126 Catania (CT)*
[c]*Department DIIES, University "Mediterranea" of Reggio Calabria, Loc. Feo di Vito, 89122 Reggio Cal. RC*
[d]*Department DICEAM, University "Mediterranea" of Reggio Calabria, Loc. Feo di Vito, 89122 Reggio Cal. RC*

Abstract

Nowadays, a challenge for the "Internet of Things" (IoT) world is represented by the necessity of facing very complex and interactive tasks, such that IoT devices have to be equipped with hardware having very powerful capabilities. All this becomes particularly critical in presence of small and low-cost IoT devices. A way to deal with such problems is represented by the possible virtualization of the IoT environments over the cloud, the so called Cloud-of-Things (CoT), and then to associate each device with one or more software agents working in the Cloud environment. Moreover, the convergence of these technologies allows IoT devices to take significant benefits also by the social attitude of software agents to interact and cooperate. In this context, based on Machine-to-Machine (M2M) interactions, the choice of the partner for cooperating is a sensitive question, particularly in open and heterogeneous environments. If an agent does not hold suitable information to carry out a reliable choice then, similarly to real communities, it can ask information to other agents it considers as trustworthy. In this context, agents cooperation must be supported by a proper trust model which helps to select potential partners. This process can be further improved by partitioning the agents in different groups based on trust relationships. This way, each agent has the possibility to prefer the interactions with the agents belonging to its group that are, from its viewpoint, the most reliable for avoiding malicious behaviours and threats of different nature. To this purpose, we designed an algorithm, named CoTAG (CoT Agent Grouping algorithm), to form agent groups on the basis of information about reliability and reputation collected by the agents. To verify the efficiency and effectiveness of this algorithm, we carried out some experimentations in a simulated scenario. The results confirm the potential advantages deriving by the adoption of our proposal.

Keywords:
Cloud of Things; Local reputation; Multiagent system; Trust system; Voting

Email addresses: giancarlo.fortino@unical.it (G. Fortino), messina@dmi.unict.it (F. Messina), domenico.rosaci@unirc.it (D. Rosaci), sarne@unirc.it (G.M.L. Sarné)

1. Introduction

In 1999 Kevin Ashton prophetically foresaw the "Internet of Things" age [1] that is connecting people and physical objects over Internet, on a massive scale[1] in an interactive manner. Internet of Things (IoT) realizes multi-dimensional and context-aware smart environments[2] for every aspect of our everyday-life [4]. In general, IoT can be assumed as a global network infrastructure composed by heterogeneous cooperating smart objects able to sense, reason, collaborate and act in real time upon the environment by using a wide range of different sensory, communication, networking and information technologies [5, 6] and capable of social interactions [7, 8].

Nowadays, a challenge for the IoT world is represented by the necessity of facing complex interactive tasks and, consequently, increasing hardware and power capabilities are required to IoT devices. This issue becomes particularly critical in presence of small and low-cost IoT devices. Therefore, given IoT potentialities, Information and Communications Technology (ICT) industries and important standard organizations are supporting IoT technical, social, and economical challenges by developing both new technologies and standards [9] to manage the complexity placed by the IoT world at best.

At the same time, Cloud Computing (CC) has emerged as a successful Internet information technology addressed to share ubiquitous, reliable, configurable and highly scalable services [10, 11], a mainstream in processing and storing data to form knowledge ubiquitously accessible in distributed environments and in an interoperable fashion [12]. In this scenario IoT and CC converged to realize the so called Cloud-of-Things [13, 14] (CoT). Such a tighter integration is strategically motived from the necessity to support, in a scalable way, the computational and storing requirements [15] coming by an overwhelming number of ubiquitous, heterogeneous and often small and low-cost IoT devices for discovering, composing or making available new services. Moreover, the virtualization of more IoT environments over a unique CC also makes easier to support mobile devices in their nomadic activities [16].

In particular, associating IoT devices with software agents, working on their behalf in the Cloud [17, 18], can provide several different benefits: first of all, software agents are able to manage complex tasks independently from the IoT hardware and power capabilities; secondly, the convergence of these technologies allows IoT devices to take significant benefits also by the social attitude of software agents to interact and cooperate, very useful in engaging IoT challenges. In other words, the level of "satisfaction" in the M2M (Machine-to-Machine) interactions occurring among devices in the composition of CoT services is highly influenced by the choice of the "partner" for cooperating [19], particularly in open and heterogeneous environments. Therefore, when an agent has not suitable information to choose a reliable partner then, similarly to real communities, it can ask information to other agents it considers as trustworthy. As a result, agent cooperation can be promoted by supporting agents with reliable recommendations about their potential partners [20]. To this purpose, the intuition underlying our proposal is that of

[1] In our context, an *object* (i.e., a thing) is a physical (or virtual) entity that throughout its lifetime is precisely traceable in space and time, sustainable, enhanceable and uniquely identifiable [2].

[2] A smart environment is characterized by the capability of acquiring knowledge about itself and its inhabitants so as to adapt itself to their needs and behaviors [3]

supporting this process by encouraging agents to form groups of reliable recommenders.

In fact, software agents (in the interest of their associated devices) can form complex social structures, as agent groups, on the basis of some type of social relationships among the group members [21, 22]. Given their relevance in real and virtual communities, the dynamics underlying formation, evolution and roles of social groups have been widely investigated in the literature [23, 24, 25, 26]. In this context, a common viewpoint considers that groups should be formed on the basis of both structural and semantic similarities (representing commonalities of relations, interests and preferences) [27]. Due to the high heterogeneity of devices, a similarity approach could be not applicable at all the devices and, therefore, different criteria have to be adopted. We observe that an important property considered in forming groups within a community is a high level of mutual trustworthiness among the group members. This is particularly important in promoting agents mutual collaboration based on their mutual trust [28]. Therefore, we consider the trust-based processes devoted to form agent groups of reliable recommenders over a CoT context as worthy of investigation because, potentially, such groups can significantly improve the devices activities.

1.1. The scenario

Basing on the premise above introduced, we consider a CoT environment where devices, heterogeneous for characteristics and/or goals, consume/produce services and/or extract/exchange knowledge by exploiting the assistance, over the cloud, of personal software agents. More formally, let us denote with A the set of software agents, associated to IoT devices; these agents, as already discussed, live in the Cloud. For sake of simplicity, the set of agents and their relationships are represented by means of a graph $G = \langle N, L \rangle$, where N represents the set of nodes belonging to G and each node $n \in N$ is associated with a unique agent $a \in A$, while L is the set of oriented links where each link $l \in L$ represents a relationship occurring between two agents. Since each IoT device and its associated agent are identified in an univocal manner, from this point we assume that the single device and its associated agent $a \in A$ are the same entity.

With respect to the agent group membership, we consider that the agents are free of belonging to one or more groups, as well as to leave a group on the basis of their convenience. At the same time, we assume that each group is coordinated by an agent group administrator that, to maximize the effectiveness of the group itself, can contact other devices (i.e., agents) to join with or to remove from the group those agents resulted ineffective.

To reach their goals, the consumer agents can exploit some data services (s) made available by other agents only for payment. Note that each agent can be a consumer or a provider of services[3]. In requiring a service to a provider, an agent might take benefit from its past experiences, but if they are not sufficient to perform a good choice it can also require the opinions of other agents [29].

More formally, we assume that the generic agent a_i has not a suitable direct past experience about a provider agent a_j, it can ask a recommendation $rec \in [0, 1] \subseteq \mathbb{R}$, where $\mathbb{R}$ is the set of

[3] In presence of an agent performing both the roles (i.e., a prosumer) by acting as provider for some services and consumers for other services, we consider these activities as disjointed.

real numbers, to another agent a_r. If a_r belongs to the same group of a_i this recommendation is provided for free, otherwise a fee has to be paid from a_j to a_r after the recommendation was provided[4]. This mechanism implies that, on the basis of trust measures, groups are interested in accepting those agents having a high reliability and helpfulness; at the same time agents are interested to be affiliated with those groups formed by agents with a high reliability and helpfulness. However, remember that all the services are provided only for payment, differently from the recommendations that could be provided also for free; in this way, the proposed scenario has a competitive nature.

In this context, to evaluate the helpfulness of an agent we consider the effectiveness of its recommendations, while trivially that of a group is the average of the helpfulness of its members. In particular, a_i assigns a feedback $f_{i,r} \in [0, 1] \subseteq \mathbb{R}$ to each agent recommended by a_r that it used, where $f_{i,r} = 1$ means that it perceived the maximum level of satisfaction and, vice versa, $f_{i,r} = 0$ will mean a null satisfaction for it. The difference between the feedback $f_{i,r}$ and the corresponding recommendation $rec_{i,r}$, in absolute value, provides a measure of the helpfulness of this recommendation; the average of the measures provided by a_r represents its helpfulness.

1.2. Our contributions

With respect to the scenario above described, our first contribution consists of exploiting trust measures to model a distributed group formation process aimed to maximize from one hand the benefits of an agent to join with a group and, from the other hand, that of a group in accepting a new member. To this aim, we designed a distributed algorithm, named CoT Agent Grouping (CoTAG) matching devices and groups to improve individual and global satisfaction[5] into the CoT on the basis of trust measures taking into account the agent helpfulness in providing useful recommendations.

In computing trust measures once the direct experience (i.e., the reliability) of an agent is inadequate to trust another agent then also the opinions coming from other agents can be exploited to build their reputation. However, in similar contexts, when it is required to calculate the reputation of an agent often a global approach (i.e., *global reputation*) is adopted by asking an opinion at all the agents belonging to A. We observe that in huge communities, as well as in large groups, this approach is unfeasible also because an agent a_i usually interacted with a limited number of other agents so that the most part of the population of A is unreferenced with respect to their trustworthiness as it is perceived by a_i. Differently, the past interactions of a_i can provide it of the necessary experience to trust its counterparts on the basis of their perceived helpfulness. We name *entourage* the set of trusted agent counterparts (i.e., friends) of a_i and identify it with E_i.

In this respect, in place of the global reputation we adopt a *local reputation* [30] approach based only on the opinions coming from the friends of an agent. Indeed, similarly to real user communities, when a user is lack of experience to reach a reliable decision then usually he/she

[4] To assure the competitiveness, each agent can satisfy at most Y requests of recommendation for payment.

[5] Note that, based on the social capital theory [21, 22], in a competitive context the main interest should be in improving the social capital of the groups (in other words, groups will be tend to be homogeneous with respect to the agent skills so that groups will tend to be also very different among them under this viewpoint); conversely, in a not competitive context the main goal should be that of improving the social capital of all the community.

will require an opinion to his/her friends. Only if these opinions are not sufficiently representative then he/she will ask an opinion also to the friends of friends and so on The benefits deriving by the adoption of a *local reputation* are particularly important in a CoT context and consist of *i*) avoiding heavy computational tasks and communication overloads to collect opinions and evaluate the trustworthiness of their sources and, at the same time, *ii*) increasing the system reactivity because only a little share of A is involved in this process.

As a consequence, each agent a_i has to maintain the information about two sets, the first one is the set E_i of *trusted agents* belonging to its entourage, useful in the group formation process, such that $E_i \subseteq A$. The other set stores the information about the *groups* $H_i = \{g_{i_1}, \dots, g_{i_k}\}$ which a_i is affiliated with[6] such that $\bigcup g_i \subseteq A$. Note that E_i and/or G_i could be empty as, for instance, for a newcomer agent.

As a second contribution, we also deal with the modality to combine trust values to take a decision about accepting or refusing an affiliation request to a group. To this aim, different techniques have been proposed in the literature but all of them diverge from similar processes having place in human societies, which mainly consist of some form of voting that is one of the most important forms to allow democratic choices in a community [31]. However, risks of manipulation are intrinsic in the voting itself, for example due to strategic voting [32], and a great attention is commonly given to ensure the correctness of the voting results. In this context, we propose to form groups in a simple way by adopting a voting mechanism, where each vote is represented by a trust value obtained by a suitable combination of reliability and local reputation.

The distributed algorithm for group formation we will describe in Section 4 is designed to lead the agents community to a configuration of groups with a high level of mutual trust among its members. Moreover, some experiments we will present in Section 5 highlight some interesting aspects related to the algorithm behavior. Indeed, if a few design parameters are properly set, our algorithm will drive a group formation by ruling out very quickly the unreliable agents from the groups. It is also interesting the observation that a different choice would be suitable to allow a few untrusted agents to continue to join with some groups, thus to obtain advantages from the services of the trusted agents within the group.

Finally, to verify efficiency and effectiveness of this algorithm, we carried out some experimentations, in a simulated agent CoT scenario, which confirmed the potential advantages deriving by the adoption of our proposal.

1.3. The plan of the paper

The rest of the paper is organized as follows. Section 2 gives an overview on the related literature. Section 3 describes the adopted local trust model and voting mechanism, while Section 4 presents the CoTAG algorithm to form groups. The experimental results are presented in Section 5 and in Section 6 some conclusions are drawn.

[6] To assure the competitiveness, each agent can belong at a fixed number groups and each group can affiliate a fixed number of agents.

2. Related Work

An important issue in open, competitive and distributed scenarios is to provide to involved actors a comfortable environment where realizing own activities. In this respect, it is important to limit the vulnerability with respect to a large number of potential threats, mainly due to malicious and/or disliked behaviors [33, 34].

To this purpose, two different and complementary approaches, namely cryptographic techniques and trust systems, can be exploited. The former is effective in fulfilling privacy and authentication issues or, in other words, against outside attacks, while trust systems excel in improving benefits or mitigate risks for unreliable partners [35] independently if they are human, objects or virtual entities [36]. Unfortunately, computational and storage capabilities, as well as power consumption, can make the adoption of cryptographic techniques unfeasible for a wide number of CoT devices [37].

Given its interdisciplinary nature, trust has been widely investigated within the literature by means of an overwhelming amount of analysis, models and architectures intersecting many areas like computer science [38], sociology [39], economy [40] and so on[7]. From a practical point of view, trust affects almost every decisional process and social interaction involving both human and virtual activities [47, 48].

More in general, in real and virtual communities, several approaches based on a trust criterion deal with the problem of suggesting *i*) to a group if accepting a candidate for the affiliation or *ii*) to a member of a community the groups which are the most suitable for joining with. These problems are more known, respectively, as *group recommendation* and *group affiliation* problems. Recently, the relevance of trust in profitably support group formation processes has been verified in [24, 26], where trust measures have allowed to form groups more stable over time with respect to those groups formed without to consider trust. As an example, in [49] a trust-based agreement procedure is adopted in a P2P system to form groups on the basis of the agent trustworthiness. This because the expectations to be engaged in satisfactory interactions is higher among the members of trust-formed groups and a group receives more benefits when trusted relationships occur among its members.

To compute trust in social communities, the most important factors to consider are the informative sources [50], the aggregation rules [51] and the modalities for inferring trust [52], but also if trust is computed in a local or a global way and by means of a centralized or a distributed approach. Some studies found that the accuracy of local trust is greater when a personal viewpoint is used [53], while the computational cost tightly depends by the horizon depth [54]. The predominance of local trust is particularly true in large communities where each actor usually interacts with a narrowest share of the community so that the most part of it is unknown and unreferenced like their opinions. For instance, in [55] the most accurate results are obtained in inferring trust values on the shorter paths (i.e., those paths more closed to the trustor).

[7] For such a reason, an overall contextualization of this paper within these backgrounds is beyond our aims so that, in this section, the examined approaches are those that, to the best of our knowledge, come closer to the matter presented in this paper. The interested reader might refer to a considerable number of surveys that investigated on the state-of-the-art, among which [41, 42, 43, 44, 45, 46].

In social contexts, trust processes occurring in a community are usually represented by a graph, named *trust network*, where nodes represent the members and oriented edges represent trust relationships (usually sparse). Different techniques use the topological properties of the trust networks as, for example, in [56] a variant of the Breadth First Search is adopted to gather the reputation scores and, by using a voting, to compute an updated reputation rate for each user, while in [57] updated trust scores are propagated only by using fixed length paths.

Local trust approaches are adopted by the TidalTrust [55], the MoleTrust [58] and the TrustWalker [59] algorithms. The first one states that the most accurate trust predictions derive from the closer neighbors, although part of the neighboring could be ignored if the trust network is too sparse. A backward exploration by fixing a maximum depth in the search-tree of the trust network is exploited by MoleTrust to calculate trust scores. In particular, the trust score at depth x is computed only by using the trust scores at depth $x - 1$. The last algorithm is aimed to suggest items and, to avoid precision loss due to data sparsity, considers ratings for the target and the similar items basing on (*i*) a random walk on the trust network by limiting the search depth and (*ii*) a probabilistic approach to select items by giving preference to the closer raters and to the items comparable with the target.

Another characteristic of our proposal is that of adopting a voting to reach a decision within a group. Voting mechanisms [60, 61], in human and virtual communities, optimize the social utility [62] and avoid conflicts [63] by giving equal dignity to different interests and opinions. However, in huge communities a global voting could be difficult, or unfeasible, to realize and in these cases a local voting might represent the best alternative [64]. Unfortunately, any "ideal" global or local voting procedure exists because all of them can be affected by manipulation, like strategic vote [65]. This aspect is very critical for software agent communities, indeed agents can efficiently and effectively examine manifold manipulation opportunities [66, 67, 68], but this problem is assumed as orthogonal with respect to the focus of our proposal where trust drives voting.

Trust and voting processes are different for characteristics and adopted models, although conceptually they are rather similar. Indeed, like a bet, a voter (i.e., a trustor) places own expectations (independently from a social or selfish nature) on the voted (i.e., on the trustee) to receive individual, or collective, benefits [69] by one or more future events or behaviors [70]. Furthermore, in a similar way, local trust and local voting fit with real or virtual communities denoted, from one hand, by a great population and, from the other hand, by poor communication infrastructure or, for some IoT devices, by computational and storage constraints. For instance, in [71] a local trust-based voting is implemented, in a mobile wireless scenario, to accept or not a node in a transmission path basing on its trustworthiness as perceived by the other nodes. The actual trust of a node is propagated by mutual acquaintance on an oriented trust network (including only those nodes placed at one hop of distance) by combining its neighbors' confidence values considered as trust measures. As a consequence, a node will be trusted/distrusted by using a voting scheme on a local network. In [72] a group affiliation procedure adopts a democratic voting procedure where each vote is driven by trust built on the basis of a local trust-engine.

Furthermore, in [73] it is proposed a framework to discovery faulting sensors by using their trustworthiness, identified by a value named *SensorRank*, modeled by a Markov chain on the sensor network. This value is used in a network voting scheme, named *TrustVoting*, where each vote implicitly represents the number of references of a node (coming from its neighboring) supporting its opinions. The voting procedure considers a locality criterion by weighting the votes in an inversely proportional way to their distance from the target node on the sensor network. Similarly, *SocialTrust* [74] is a framework to realize users' social trust groups by adopting a relationship model, where the size of each group can range from the direct user's neighboring, plus the neighbors of his/her neighbors, and so on. By limiting the radius it is possible to limit the impact of malicious users. In *SocialTrust* each user rates each other user he/she interacted and converts such rates in a vote (e.g., *good* or *bad*). Three possible voting mechanisms, at increasing levels of security and resilience, are made available.

Finally, we present some proposals of trust systems appositively conceived for IoT and CC contexts[8]. Indeed, in recent years, researchers have begun to give attention to the peculiarities of these environments by developing specific techniques and strategies [37, 75, 76, 77, 78]. For instance, the proposal presented in [79] assumes that two interacting IoT devices can mutually trust the other and propagating, with a *word of mouth* mechanism, their trust evaluations about the other nodes, like suggestions. In [80] a social IoT scenario behavioral information are used to limit malicious behaviors by using a subjective trust system where each node evaluates the trustworthiness of its friend nodes by considering the direct experiences and the opinion of the common friends. In the same social context, Chen et al. proposed in [81] a trust system to take into account the dynamic evolution of social relationships by conceiving a trust-based service management able to adapt itself to the unavoidable trust fluctuations. In [82] it is assumed that the IoT devices identities are unknown in advance, so that trust among two devices is based on the past direct interactions. The devices privilege to exchange services and resources with devices they already trusted and, in such way, reinforcing such relationships.

In the CC area, different authors have dealt with the problem of choosing a trustworthy CC provider without to rely only on the Service Level Agreements (SLAs) which, for instance, could be not consistent. For example, *i*) in [83] a multi-faceted Trust Management system for a CC marketplace is proposed for evaluating the CC providers trustworthiness in terms of more attributes (e.g., security, performance, compliance) assessed by different sources and trust information, while *ii*) in [84] is designed a trust management architecture for CC marketplaces capable to support customers in identifying trustworthy CC providers by verifying suspicious feedbacks arising by system and social threats. CC federation are considered in [29], where a fully decentralized trust-based model for large-scale federations is designed to allow any node to find the most suitable collaborators in an efficient way, avoiding exploration of the whole node space. More specifically, the nodes of a federation are organized in an overlay network on the basis of suitable criteria, by also including trustworthiness, to identify a suitable set of candidate nodes.

[8] From a trust viewpoint, the CC and the IoT devices composing an integrated CoT environment are substantially independent from a functional viewpoint.

3. The Trust Model

In this section, we describe the trust model which supports the group formation procedure discussed in section 4.

3.1. The Local Trust model

The *local trust* is a measure which takes into account reliability, local reputation, and helpfulness. All the three measures range in the real interval $[0, 1]$, where 0/1 represents their minimum/maximum values. We assume that trust relationships represent a network, i.e. a graph on which a directed edge between two nodes (i.e., agents) represents the level of trust that an agent has in another agent. In this context, the ego-network of an agent $a_i \in A$ is represented by the sub-graph $E_i \subseteq G$, formed by those nodes (i.e., agents) connected to a_i by oriented paths included in a fixed horizon. For example, Figure 1-A represents a small virtual community of eight agents, and Figure 1-B represents the ego-networks of the agent a including all the nodes of the virtual community for which a path to a there exists. In particular, while some agents are directly connected to the agent a by a path of length 1 (red colored links), some other agents are indirectly connected to a by a path of length 2 (blue colored links).

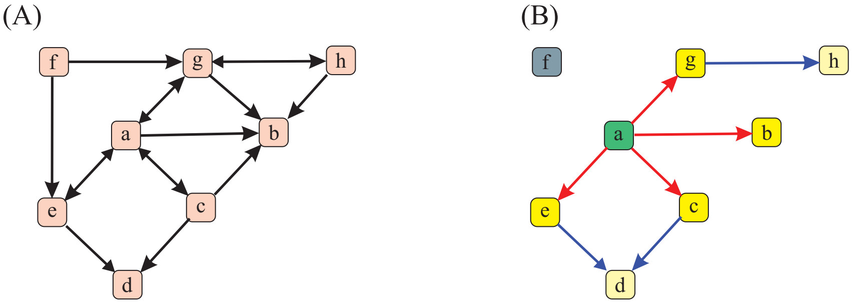

Figure 1: A) - An example of virtual community; B) - The ego-networks of the agent a connected by paths formed by 1 link (e.g., all the agent connected to a by red links) and by 2 links (e.g., all the agents connected to a by red and blue links).

For a pair of generic nodes $i, j \in G$ (i.e., the associate agents a_i and a_j), the local trust that i has about j (i.e., $\tau_{i,j}$) is defined as a combination of the reliability $\rho_{i,j}$ and the *local reputation*, denoted as $\sigma_{i,j}$.

The reliability ρ_{ij} represents a measure about the confidence that the agent a_i has about the capability, at a certain time, of the agent a_j of providing good suggestions. By observing that trust is an asymmetric measure since, generally, $\rho_{i,j} \neq \rho_{j,i}$, the measure ρ_{ij} is computed by averaging all the q feedbacks $f_{i,j} \in [0, 1] \in \mathbb{R}$ that a_i assigned to a_j at the end of the q interactions carried out with it. More formally:

$$\rho_{i,j} = \frac{1}{q} \cdot \sum_{k=1}^{q} f_{i,j}^{k} \tag{1}$$

The *local reputation* $\sigma_{i,j}$ measures how much, on average, the agents belonging to the ego-network of a_i (i.e., E_i), estimate the capability of a_j of having good interactions. To this purpose, let $rec_{r,j} \in [0,1]$ be the recommendation provided by an agent a_r about another agent a_j and let $\epsilon_{i,r} \in [0,1]$ be the (average) helpfulness of this agent a_r (in providing reliable suggestions) as it is perceived by the agent a_i on the basis of its experience, represented by the feedbacks it released after having interacted with the agent recommended by a_r (obviously, if any recommendation was provided by a_r to a_i, then the helpfulness of a_r perceived by a_i will be null, i.e. $\epsilon_{i,r} = 0$). Therefore, by considering that, at a certain time, the agent a_r provided m recommendations to a_i about other agents, the helpfulness $\epsilon_{i,r}$ of a_r perceived by a_i is computed, with respect to all the accepted suggestions coming by a_r, as:

$$\epsilon_{i,r} = \frac{1}{m} \cdot \sum_{s=1}^{m} |f_s - rec_s| \tag{2}$$

We define a second weight denoted as ω which considers the distance between a_i and the recommender agent a_r belonging to the ego-network E_i, in order to give less importance to those recommender agents which are more "far" from a_i. Now, let $\widehat{l}_{(i,r)}$ be the shortest path between a_i and the generic recommender agent a_r, then $\omega_{i,r}$ is computed as:

$$\omega_{i,r} = 2^{-(\ \widehat{l}_{(i,r)}-1)} \tag{3}$$

Therefore, by assuming that a_i is able to exploit a number p of recommenders in its ego-network to receive recommendations about a_j, then $\sigma_{i,j}$ is calculated as:

$$\sigma_{i,j} = \frac{1}{p} \cdot \sum_{r=1}^{p} \big(\epsilon_{i,r} \cdot \omega_{i,r} \cdot rec_{r,j}\big) \tag{4}$$

Finally, we can compute the trust measure that an agent a_i has about an agent a_j by combining reliability and local reputation (which also takes into account the helpfulness) as:

$$\tau_{i,j} = \alpha_i \cdot \rho_{i,j} + (1 - \alpha_i) \cdot \beta_{i,j} \cdot \sigma_{i,j} \tag{5}$$

where α and β are two parameters ranging in $[0,1] \in \mathbb{R}$. The parameter α simply weights reliability and local reputation for giving more or less relevance to one or other. The parameter β is computed as $\beta_{i,j} = p/\|E_i(x)\|$ and takes into account the dependability of $\sigma_{i,j}$ on the number of p nodes belonging to E_i that contributed to compute $\sigma_{i,j}$ (this because if the number of these nodes is too small then a_i will not receive a sufficient amount of information about a_j from its ego-network and the local reputation measure loses of relevance). Furthermore, note that to tune the local reputation, in Equation 5 the agent opinions are weighted by using these two parameters because, as highlighted in [42], the capability to provide reliable opinions is unrelated to other aspects. Moreover, note that in presence of a newcomer agent then suitable "cold start" values of reliability, reputation and helpfulness are adopted.

The "trustworthiness" of a group g, as perceived by a_i (i.e. $\tau_{i,g}$), can be determined by simply averaging all the trust measures computed by a_i for all the agents belonging to g. Similarly, the "trustworthiness" of an agent a_i, as perceived by a group g (i.e. $\tau_{g,i}$), can be determined by simply averaging all the trust measures about a_i computed by all the agents belonging to g.

3.2. The Voting Mechanism

In this section we introduce a simple voting mechanism [85], which is based on the local trust measure defined in the previous section 3.1.

Each time a decision about a new membership with a group g has to be taken, all the agents belonging to g will express a preference (i.e., a vote) $v \in \{0, 1\}$ to accept or not this agent into g (e.g., 0/1 means "not accept"/"accept"). The vote will depend from *i*) the local trust measure that the voter computed about the candidate, also on the basis of the recommendations coming from its ego-network and *ii*) a suitable threshold $\Gamma_g \in [0, 1]$:

$$v = \begin{cases} 0 & \textit{if}\ \tau < \Gamma_g \\ \\ 1 & \textit{if}\ \tau \geq \Gamma_g \end{cases} \tag{6}$$

In the following, we represent the voting process referred to a group g for a potential new member y by adopting the voting criterion v (e.g., see formula 6), as the output of a function $V(g, v, y)$. For instance, a reasonable strategy may be that of accepting a requester into a group only if the majority of the members of g vote for its acceptance.

4. The Distributed CoT Agent Grouping Algorithm

In this section, we describe the distributed CoT Agent Grouping algorithm (shortly CoTAG), which is designed to maximize the benefits of the single agent to join with a group and, at the same time, the benefits of a whole group when it decides to accept a new member. The algorithm consists into two procedures: the former, illustrated in Algorithm 1, is designed to be executed by each single agent of the CoT environment, while the latter, illustrated in Algorithm 2, is designed to be executed by each agent that acts as group *administrator*. As we explain later in Section 4.1, the aim of Algorithm 1 is to allow each agent of the CoT environment to find the "best" groups to join with, in terms of average value of trust $\tau_{i,g}$, where g denotes the generic group. Conversely, as we explain later in Section 4.2, the aim of Algorithm 2 is to allow every group administrator to evaluate the possible joining of a new member with the group itself. Also in this case, the evaluation is based on the mutual trust among members of the group and the potential new member.

The reader may also refer Table 1 for the symbols used in the description of the CoTAG procedures, explained in sections 4.1 and 4.2.

4.1. CoTAG: the procedure executed by every agent.

This procedure is represented by Algorithm 1, where $H_i \subset Gr$ is the set of the groups to which the agent a_i is affiliated to and W is a parameter representing the maximum number of groups that an agent can join with, while M represents the maximum number of groups the generic agent is capable to analyze. Furthermore, we suppose that the generic agent a_i stores the local trust measure $\tau_{i,g}$ of each group $g \in H_i \subset Gr$ contacted in the past and the time $\hat{t}_g$ elapsed from the last computation. Moreover, let π_i be a time threshold fixed by the agent a_i, and $\theta_i \in [0, 1]$ a threshold on the trust value between the agent a_i and the generic group $g \in H_i$. This

Table 1: Table of the main symbols

Symbol	Description
A	set of agents associated to the IoT devices
G	graph representing the set of agents and their relationships $G = \langle N, L\rangle$
E_i	set of agents belonging to the ego-network of an agent a_i, with $E \subseteq G$
Gr	set of all the groups
H_i	set of the groups which an agent a_i is affiliated, with $H_i = \bigcup g_i \subseteq A$
K_g	set of agents affiliated with a group g
M	maximum number of new groups the single agent is able to analyze
W	maximum number of groups that an agent can join with
R	maximum number of agents belonging to a group
S_c	set of candidate groups
$V(\cdot)$	voting function
Y	set of groups randomly chosen, with $\|Y\| \leq M$
a	agent
a_g	agent administrator of the group g
g	generic group
$\tilde{t}$	time elapsed from the last execution of the CoTAG-A procedure for an agent
$\hat{t}$	time elapsed from the last execution of the CoTAG-B procedure for a group
θ	threshold on the level of trust between an agent and a generic group
ϕ	time threshold fixed by the agent administrator of a group
π	time threshold fixed by an agent
τ	trust

procedure is executed by the agent a_i to improve its own "configuration" of groups in terms of overall mutual trust with the related peers.

First of all, values of the local trust $\tau_{i,g}$ are calculated whenever the values previously stored are older than the threshold π_i (lines 1-3). Then, a set of candidate groups S_c, with $\|S_c\| < W$, is built on the basis of the local trust of the groups, and sorted in decreasing order, based on the trust values $\tau_{i,g}$, while Y is a set of groups randomly chosen and, finally, the set Z is given by union of Y and H.

As described in the definition of the set Y and Z, the set S_c might contain some groups already belonging to the existing set H_i, while some others might be new groups that were selected at random and put into the set Y. Then, based on the groups in the set S_c which are not in the set H_i, the agent a_i will be able to improve the quality of its choices by joining with those groups. The two loops in lines 6-18 represents the kernel of the procedure, after that $H_i = S_c$.

4.2. CoTAG: the procedure executed by the agent administrator of every group.

This second procedure is designed to be executed by the administrator of every group in the CoT environment. It is represented by the pseudocode listed in Algorithm 2. Let $K_g \subset Gr$ be the set of the agents affiliated to the group g, where $\|K\| \leq R$, being R the maximum number of agents allowed to be affiliated with the group g, while the set X is the union of the set K_g with

Algorithm 1 CoTAG: the procedure executed by each agent a_i in the CoT environment

```
Input:
  $H_i \subset Gr, W, \pi_i, \theta_i$;
  $Y = \{g \in G\}$ a set of groups randomly selected : $\|Y\| = M \leq W$, $H_i \bigcap Y = \{\ \}$,   $Z = (H_i \bigcup Y)$

 1: for $g \in Z : \hat{t}_g > \pi_i$ do
 2:     Compute $\tau_{i,g}$ by exploiting the agents belonging to $g$.
 3: end for
 4: $m \leftarrow 0$
 5: Let be $S_c = \{g \in Z : \tau_{i,g} \geq \theta_i\}$, with $\|S_c\| = W$
 6: for all $g \in S_c : g \notin H_i$ do
 7:     send a join request to the agent administrator of $g$
 8:     if $g$ accepts the request then
 9:         $m \ \leftarrow \ m + 1$
10:     end if
11: end for
12: for all $g \in H_i : g \notin S_c$ do
13:     Sends a leave message to $g$
14:     $m \leftarrow m - 1$
15:     if (m==0) then
16:         break
17:     end if
18: end for
```

the agent a_i candidate to be affiliated with the group g. We assume that the agent administrator a_g of a group g stores the values of the local trust computed by the members of its group for the agent a_i which has sent the request to join with, and the timestamp $\tilde{t_i}$ of its retrieval. Moreover, let ϕ a time threshold fixed by the administrator a_g.

The procedure performed by the group agent administrator a_g is executed by the administrator of the group once an agent, denoted as a_i, sends a join request to a_g. By lines $1 - 5$, the administrator a_g asks to the members of its group the updated local trust values about the agent a_i. At this case, two different situations might occur:

1. the size of X has not reached the maximum R (line 6), then all the agents in g are asked to provide a personal preference (i.e. a vote). All the preferences are then combined by means of the function $V(\cdot)$ defined in Section 3.2. Then, agent a_i is admitted or not in the group on the basis of the result of the function $V(\cdot)$.
2. the size of the set X has reached the maximum R. In this case, a simple binary preference, e.g. that given by the function $V(\cdot)$ will not be enough. Indeed, since the size of the set X has reached the maximum allowed, if the agent a_i is admitted into the group, another agent has to be rejected from the group. As a consequence, the agents must be associated to a rank in order to be comparable with its peers. In this case, a natural measure to rank the agents is the trust of the group vs the agent itself, which is computed as explained in

Section 3 (line 16). In particular, $\tau_{g,n}$ denotes the current value of trust between the group g and the agent $k_n \in X \bigcup \{a_i\}$.

While the first scenario is dealt with in lines $6 - 11$ of Algorithm 2, the second one is dealt with into lines $12 - 24$.

5. Experiments

In this section we report and discuss the results of a number of experiments aimed at verifying the effectiveness of the approach. The goal of our experiments is to test whether the CoTAG algorithm is able to produce a group formation with values of mutual trust within groups that, in average, are higher than those reached for different composition of groups (e.g. random or based on services features). To this end, section 5.1 describes the setting used for the experiments, Section 5.2 presents and discusses the experimental results, while section 5.3 further discusses the results.

Algorithm 2 CoTAG algorithm: the procedure executed by the administrator of every group g.

```
Input:  $K_g, R, a_i, \phi, X = K_g \bigcup \{a_i\}$;
 1: for all $k \in K_g$ do
 2:     if $\tilde{t_i} \geq \phi$ then
 3:         ask to $k$ for the updated local trust values about $a_i$
 4:     end if
 5: end for
 6: if $\|X\| < R$ then
 7:     if $V(g, v, a_i) == 1$ then
 8:         Send an accept message to $a_i$
 9:     else
10:         Send a reject message to $a_i$
11:     end if
12: else
13:     for all $k \in X$ do
14:         compute $\tau_{k,a_i}$
15:     end for
16:     Let $X' = \{k_1, k_2, \ldots, k_{\|K_g\|+1}\}$ with $k_i \in X \bigcup \{a_i\}$, ordered by trust with $\tau_{g,m} \geq \tau_{g,n}$ iff $m < n$
17:     if $X[\|K_g\| + 1] == a_i$ then
18:         Send a reject message to $a_i$
19:     else
20:         Send a leave message to the node $X[\|K_g\| + 1]$
21:         Send an accept message to $a_i$
22:     end if
23: end if
```

5.1. Experimental settings

We generated a network of 1000 different agents, assuming that they live in a virtualized environment provided by the Cloud. We remember that, in our model, each agent represents a single device living in the CoT environment. Before the simulation, the network is constructed by randomly generating 1000 trust relationships among agents, in order to build an initial network. Their values were generated on the basis of the characteristics of the agents of the community, as specified below. In particular, the ratio between trusted and untrusted agents was set as 0.5, and the values of trust were generated on the basis of the normal distribution with the parameters shown in Table 2. In this way, the initial network is very sparse, which is the desired property. Thereafter, during the simulation, the trust network will grow due to the diffusion of fresh reliability information. Moreover, before the first step of the simulation, all the groups (their number is indicated as $|Gr|$) of the system were set by randomly placing the agents into them.

Once the simulation started, at each time step a certain number of interactions among a subset of the agents was simulated, that will cause the release of feedbacks representing the "quality" of interactions among agents. Interactions were simulated by setting the expected number of generated feedbacks per step (see Formula 1), which was sampled by means of the Poisson distribution with expected value $\lambda = 50$ (see Figure 2-a). Values of feedbacks were generated by means of a normal distribution with different parameters (see Figure 2-b): for unreliable agents we set $mean = 0.2$ and $stdDev = 0.1$ – these are labeled, in Figure 2, LP (Low Performance) – while the feedback released to the reliable agents we set $mean = 0.9$ and $stdDev = 0.1$, the latter are labeled, in Figure 2, "LP" (High Performance). Finally, the ratio between reliable and unreliable agents was set as 0.5. The configuration of the remaining parameters is shown into Table 2.

Then, each step of the simulation was performed as follows:

1. a number of interactions is simulated among agents, where the number of interactions and feedback values are set as discussed in the previous paragraph;
2. 100 execution of the algorithm are simulated by triggering the agent-side of the CoTAG algorithm on 100 different agents randomly selected. In particular, each time the request of an agent to join with a group is simulated, the administrator-side of the algorithm is executed to decide whether or not to accept the agent itself;
3. a few statistics are computed.

5.2. Results

First of all, we defined the *Average Mutual Trust* among the components of a group g as $AMT_g = 1/(2\|g\|)\sum_{\substack{i,j=1\\ i\neq j}}^{\|g\|}(\tau_{i,j} + \tau_{j,i})$. Moreover, we defined the *Mean Average Mutual Trust*, for a certain configuration at a certain time-step, as $MAMT(Gr) = 1/(\|Gr\|)\sum_{i=1}^{\|Gr\|} AMT_{g_i}$.

The first set of results is shown in Figure 3-a and 3-b. Collected data span across one hundred steps of simulation. Both figures report the median value of MAMT measured after each single step of the simulation for the different values of $M = [5\div 10]$. Figure 3-b reports the first 30 steps of the simulation, while Figure 3-a reports the results obtained until 100 steps of execution of the simulations. Figure 3-a shows the slow convergence of the MAMT values –toward its asymptotic

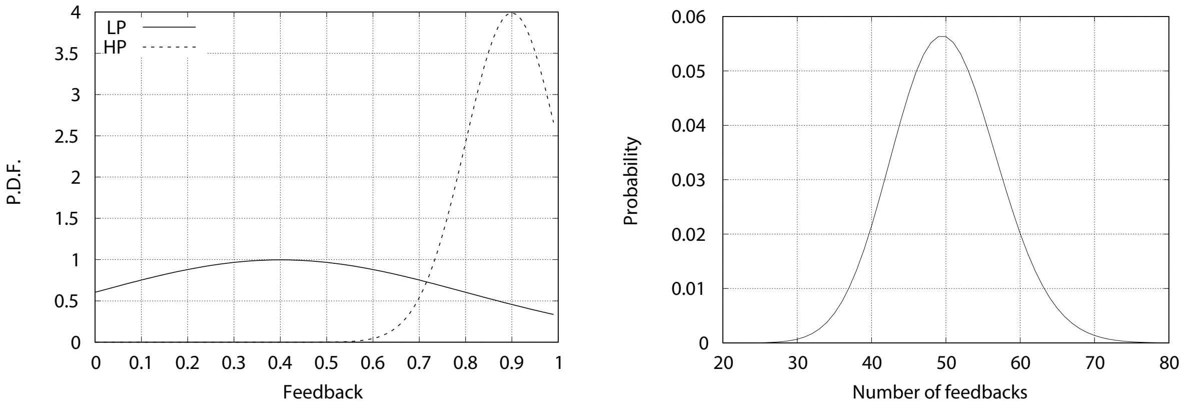

Figure 2: a) (Left) PDF of generated feedback values; b) (Right) Distribution of the number of per step generated feedbacks

Table 2: Experimental Settings

Parameter	**Setting**	**Value**
General		
No. of Agents ($\|\|A\|\|$)	Fixed	1000
No. of Feedbacks per step	Poisson distribution	$\lambda = 50$
Agents Performance (Reliability and trust)		
Low Performance (LP)	Normal Distribution	$mean = 0.9$
		$stdDev = 0.1$
High Performance (HP)	Normal Distribution	$mean = 0.2$
Cold start value of trust	Fixed	0.5
		$stdDev = 0.1$
Ratio of reliable/unreliable agents (LP)	Fixed	$0.5/0.5$
Group formation		
K (Max no. of agents per group)	Fixed	20
M (Max no. of new groups an agent is able to analyze)	Fixed	$\{5, 10, 15, 20\}$
$\|\|Gr\|\|$ (No. of groups)	Fixed	50
l_{Max} (Maximum distance of recommenders)	Range	$\{$1,2$\}$
θ (Minimum value of trust for a group to be selected as candidate for group formation)	Fixed	0.2

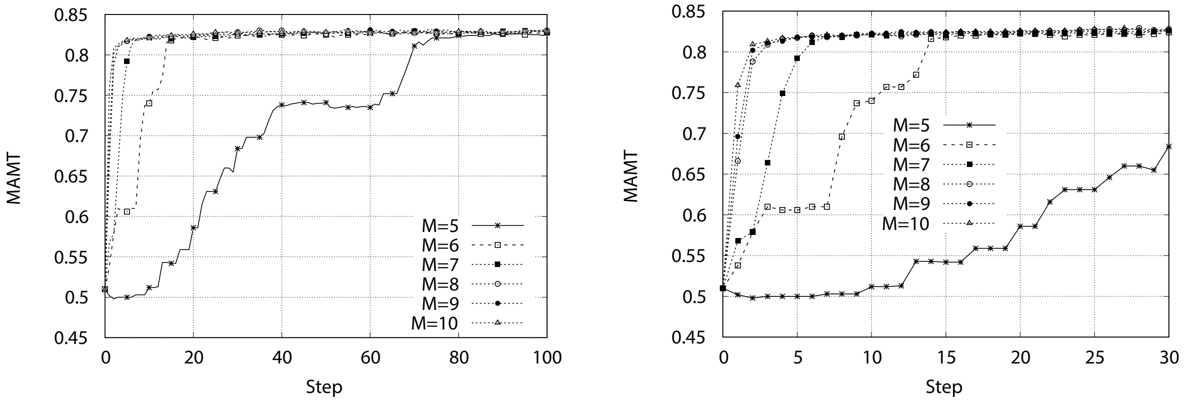

Figure 3: a) (Left) MAMT, results until 100 Steps; b) (Right) MAMT, results until 30 Steps

value–when $M = 5$. Conversely, for $M = 6$ and, even better $M = 7$, the convergence is quite fast. Details can be observed in Figure 3-b (steps $0 - 30$ of the simulation) which shows a very slow increment of the MAMT for $M = 5$ during the first 30 steps, and a radical change of behavior for $M = 6$.

The behavior described above can be explained by the fact that parameter M represents the number of new groups analyzed by the single agent a_i in the execution of Algorithm 1, which are then mixed with groups already in the set H_i in the new set S_c. Therefore, the higher the parameter M, the higher the number of new groups analyzed in the algorithm shown in Figure 1, the higher the probability to join with a new group containing untrusted agents. In this latter case, the group administrator will accept the trusted agent with high probability (since the trust $\tau_{g,i}$ is high with high probability) and will reject the agent showing the worst value of trust. As a consequence, this process will lead to an increment of the values of MAMT: the untrusted agents, sooner or later, will leave groups because they are replaced by trusted agents, which is the desired behavior.

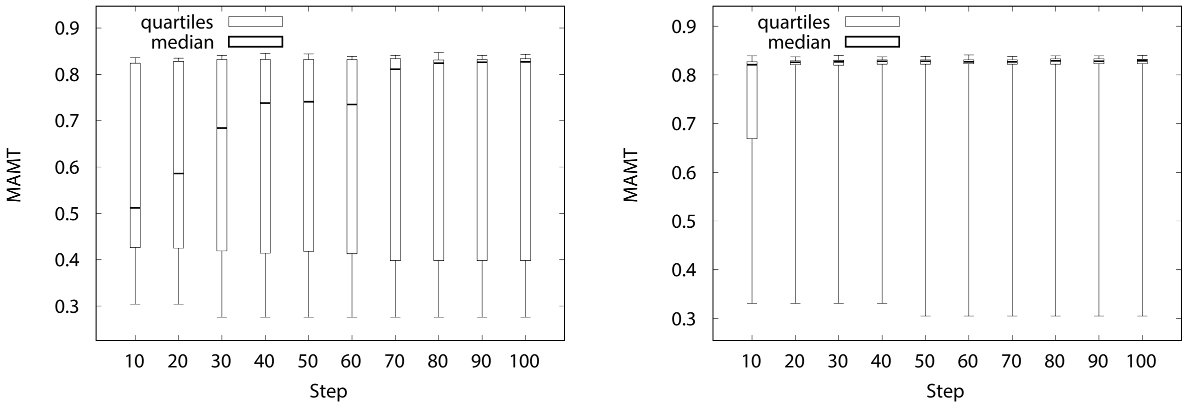

Figure 4: a) (Left) Five-number summary of MAMT, $M = 5$; b) (Right) Five-number summary of MAMT $M = 10$

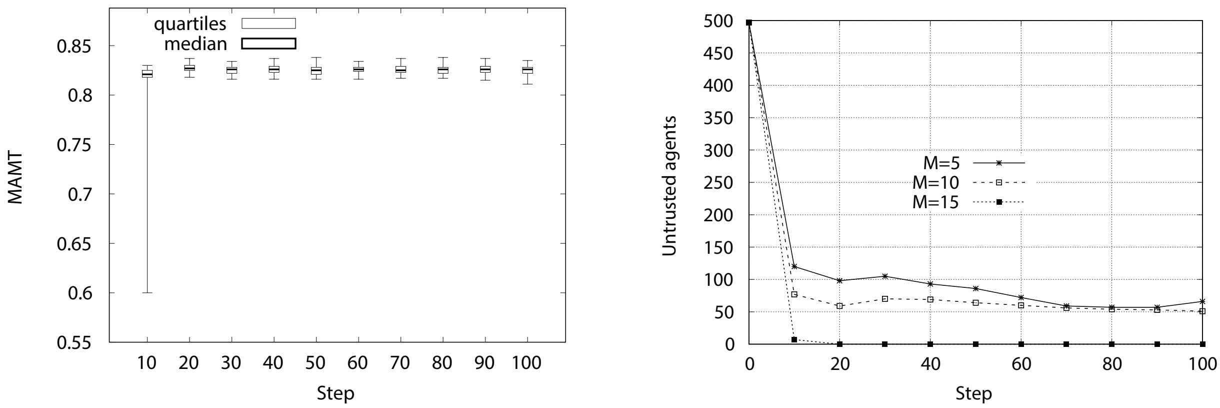

Figure 5: a) Five-number summary of MAMT, $M = 15$; b) (Right) Sum of untrusted agents vs number of steps of the simulation

To catch more details about the process explained above, we report, in Figure 4, the five-number summary of the MAMT values for $M = 5$ (Figure 4-a) and $M = 10$ (Figure 4-b). By these plots we can observe that, for $M = 5$, the MAMT minimum value and the 1st quartile are very low. These results are explained by the existence of a few groups that still host one or more bad agents. Indeed, in the execution of Algorithm 1 (executed by the generic agent a_i) the groups for which $\tau_{g,i}$ is under a certain threshold θ, are not included in the set of candidates S_c. Moreover, the set S_c is sorted by decreasing values of trust $\tau_{i,g}$. In particular, in our experiments, although the threshold θ is quite small (see Table 2), small values of M will lead to the preservation of a few groups with a few bad agents, as a consequence we will find some small values of AMT. This result is quite evident when $M = 5$ (Figure 4-a). Conversely, for large values of M, this behavior is almost negligible (Figure 4-b), although it does not disappear. To this end, we show the same plot for $M = 15$ in Figure 5-a on which is clear that, after 20 steps, untrusted agents are no longer part of any group of $\|Gr\|$.

To provide more insight about this result, we traced the composition of the groups, in term of untrusted/trusted agents, and we calculated the sum of the untrusted agents in all the groups at the simulation steps that are multiple of 10 $\{0, 10, \ldots, 100\}$, and the different values of M in the set $\{5, 10, 15\}$. we report the related plot in Figure 5-b. Results confirm the hypothesis above, i.e. when the value of M is larger than 10, almost untrusted agents are ruled out from groups, and they are replaced by trusted agents.

5.3. Discussion

In the light of the experimental results discussed in the previous section, it can be stated that the execution of the distributed algorithm for group formation called CoTAG – detailed in Section 4 – leads to a configuration of groups with a high level of (average) mutual trust among its members. In particular, the analysis of the experimental results has highlighted that, in a simulated environment, the convergence of the algorithm towards a group configuration with trusted agents is very fast if the algorithm parameters are properly set. Our analysis allowed us to clarify some interesting aspects related to the algorithm behavior. Indeed, if a few design parameters are properly set (e.g. parameter M), the algorithm will drive a group formation by ruling out the unreliable agents from the groups very quickly. Conversely, a different choice would be suitable to allow a few untrusted agents to continue to join with some groups, so they can get benefit from the services of the trusted agents within the group.

6. Conclusions

In this paper we described a CoT scenario to support the complex activities required to the CoT devices by virtualizing their physical environments over the cloud where each device can be associated with one or more software agents working in its behalf. In this way, it is possible to exploit the social attitude of software agents to interact and cooperate in an easy way as well as to form complex agent social structures, as groups. However, a satisfactory interaction tightly depends from the choice of the counterpart and when there are not sufficient information to carry out this choice autonomously, suitable recommendations can be asked to those agents considered mostly trustworthy.

To support the agents activities of their CoT devices, we propose to encourage agents to form groups of reliable recommenders by promoting a framework presenting both competition and cooperation traits. To this aim, we designed the distributed algorithm CoTAG just to form agent groups having a high mutual trust among their members. This algorithm is able to guide the group formation processes on the basis of *i*) reliability, local reputation and helpfulness measures about the agent capability of providing useful recommendations and *ii*) with a proper voting procedure. In particular, the adoption of *local reputation* measures *i*) avoids to carry out the heavy computational tasks and communication overhead required from a *global reputation* mechanism and *ii*) increases the system reactivity given that only a little share of the agent community is involved in this process.

To verify efficiency and effectiveness of this algorithm, we carried out some experimentations, in a simulated agent CoT scenario, which confirmed the potential advantages deriving by the adoption of our proposal in improving individual and group satisfaction in terms of mutual trust (i.e., reliability, local reputation and helpfulness).

In our ongoing researches, we aim at studying how the CoT agent framework can be integrated to provide a proper support in the Big Data [86], in particular the extraction of useful knowledge from Big Data sources.

References

[1] K. Ashton, That' internet of things' thing. rfid journal, 22 june 2009 (2009).

[2] B. Sterling, Shaping things (mediaworks pamphlets).

[3] J. C. Augusto, V. Callaghan, D. Cook, A. Kameas, I. Satoh, Intelligent environments: a manifesto, Human-Centric Computing and Information Sciences 3 (1) (2013) 12.

[4] N. Wu, Z. Li, K. Barkaoui, X. Li, T. Murata, M. Zhou, Iot-based smart and complex systems: a guest editorial report, IEEE/CAA Journal of Automatica Sinica 5 (1) (2018) 69–73.

[5] L. Da Xu, W. He, S. Li, Internet of things in industries: A survey, IEEE Transactions on industrial informatics 10 (4) (2014) 2233–2243.

[6] K. Ding, P. Jiang, Rfid-based production data analysis in an iot-enabled smart job-shop, IEEE/CAA Journal of Automatica Sinica 5 (1) (2018) 128–138.

[7] G. Kortuem, F. Kawsar, V. Sundramoorthy, D. Fitton, Smart objects as building blocks for the internet of things, IEEE Internet Computing 14 (1) (2010) 44–51.

[8] G. Fortino, W. Russo, C. Savaglio, W. Shen, M. Zhou, Agent-oriented cooperative smart objects: From iot system design to implementation, IEEE Transactions on Systems, Man, and Cybernetics: Systems - in press (2018). DOI:10.1109TSMC.2017.2780618.

[9] Z. Sheng, S. Yang, Y. Yu, A. Vasilakos, J. Mccann, K. Leung, A survey on the ietf protocol suite for the internet of things: Standards, challenges, and opportunities, IEEE Wireless Communications 20 (6) (2013) 91–98.

[10] B. P. Rimal, E. Choi, I. Lumb, A taxonomy and survey of cloud computing systems, in: INC, IMS and IDC, 2009. NCM'09. Fifth International Joint Conference on, Ieee, 2009, pp. 44–51.

[11] M. H. Ghahramani, M. Zhou, C. T. Hon, Toward cloud computing qos architecture: Analysis of cloud systems and cloud services, IEEE/CAA Journal of Automatica Sinica 4 (1) (2017) 6–18.

[12] Y. Wei, M. B. Blake, Service-oriented computing and cloud computing: Challenges and opportunities, IEEE Internet Computing 14 (6) (2010) 72–75.

[13] M. Aazam, I. Khan, A. A. Alsaffar, E.-N. Huh, Cloud of things: Integrating internet of things and cloud computing and the issues involved, in: Applied Sciences and Technology (IBCAST), 2014 11th International Bhurban Conference on, IEEE, 2014, pp. 414–419.

[14] P. Parwekar, From internet of things towards cloud of things, in: 2011 2nd International Conference on Computer and Communication Technology (ICCCT-2011), 2011, pp. 329–333. doi:10.1109/ICCCT.2011.6075156.

[15] A. Botta, W. De Donato, V. Persico, A. Pescapé, Integration of cloud computing and internet of things: a survey, Future Generation Computer Systems 56 (2016) 684–700.

[16] G. Aloi, G. Caliciuri, G. Fortino, R. Gravina, P. Pace, W. Russo, C. Savaglio, Enabling iot interoperability through opportunistic smartphone-based mobile gateways, Journal of Network and Computer Applications 81 (2017) 74–84.

[17] D. Uckelmann, M. Harrison, F. Michahelles, An architectural approach towards the future internet of things, in: Architecting the internet of things, Springer, 2011, pp. 1–24.

[18] G. Fortino, R. Gravina, W. Russo, C. Savaglio, Modeling and simulating internet-of-things systems: A hybrid agent-oriented approach, Computing in Science & Engineering 19 (5) (2017) 68–76.

[19] H. Zhu, M. Zhou, Role-based collaboration and its kernel mechanisms, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 36 (4) (2006) 578–589.

[20] R. Falcone, A. Sapienza, C. Castelfranchi, Recommendation of categories in an agents world: The role of (not) local communicative environments, in: Privacy, Security and Trust (PST), 2015 13th Annual Conference on, IEEE, 2015, pp. 7–13.

[21] A. Blanchard, T. Horan, Virtual communities and social capital, in: Knowledge and social capital, Elsevier, 2000, pp. 159–178.

[22] C.-M. Chiu, M.-H. Hsu, E. T. Wang, Understanding knowledge sharing in virtual communities: An integration of social capital and social cognitive theories, Decision support systems 42 (3) (2006) 1872–1888.

[23] H. Zhu, M. Zhou, R. Alkins, Group role assignment via a kuhn–munkres algorithm-based solution, IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans 42 (3) (2012) 739–750.

[24] P. De Meo, E. Ferrara, D. Rosaci, G. M. L. Sarnè, Trust and compactness in social network groups, ACM Transactions on Cybernetics 45 (2) (2015) 205–2016.

[25] H. Zhu, Avoiding conflicts by group role assignment, IEEE Transactions on Systems, Man, and Cybernetics: Systems 46 (4) (2016) 535–547.

[26] P. De Meo, F. Messina, D. Rosaci, G. M. L. Sarné, Forming time-stable homogeneous groups into online social networks, Information Sciences 414 (2017) 117–132.

[27] J. Doodson, J. Gavin, R. Joiner, Getting acquainted with groups and individuals: Information seeking, social uncertainty and social network sites., in: ICWSM, 2013.

[28] L. Fotia, F. Messina, D. Rosaci, G. M. L. Sarné, Using local trust for forming cohesive social structures in virtual communities, The Computer Journal 60 (11) (2017) 1717–1727.

[29] F. Messina, G. Pappalardo, D. Rosaci, C. Santoro, G. M. L. Sarné, A trust-aware, self-organizing system for large-scale federations of utility computing infrastructures, Future Generation Computer Systems 56 (2016) 77–94.

[30] P. De Meo, F. Messina, D. Rosaci, G. M. L. Sarné, Recommending users in social networks by integrating local and global reputation, in: Proc. of the 7th Int. Conf. on Internet and Distributed Information Systems, Vol. 8729 of LNCS, Springer, 2014, pp. 437–446.

[31] D. M. Kilgour, C. Eden, Handbook of group decision and negotiation, Vol. 4, Springer Science & Business Media, 2010.

[32] P. Fishburn, Arrow's impossibility theorem: Concise proof and infinite voters, J. of Economic Theory 2 (1) (1970) 103–106.

[33] R. Roman, J. Zhou, J. Lopez, On the features and challenges of security and privacy in distributed internet of things, Computer Networks 57 (10) (2013) 2266–2279.

[34] S. Chen, X. Chen, Z. Pei, X. Zhang, H. Fang, Distributed filtering algorithm based on tunable weights under untrustworthy dynamics, IEEE/CAA Journal of Automatica Sinica 3 (2) (2016) 225–232.

[35] J. Fogel, E. Nehmad, Internet social network communities: Risk taking, trust, and privacy concerns, Computers in Human Behavior 25 (1) (2009) 153–160.

[36] G. Han, J. Jiang, L. Shu, J. Niu, H.-C. Chao, Management and applications of trust in wireless sensor networks: A survey, Journal of Computer and System Sciences 80 (3) (2014) 602–617.

[37] Z. Yan, P. Zhang, A. V. Vasilakos, A survey on trust management for internet of things, Journal of network and computer applications 42 (2014) 120–134.

[38] T. French, N. Bessis, F. Xhafa, C. Maple, Towards a corporate governance trust agent scoring model for collaborative virtual organisations, International Journal of Grid and Utility Computing 2 (2) (2011) 98–108.

[39] T.-Y. Hsu, A. D. Kshemkalyani, Variable social vector clocks for exploring user interactions in social communication networks, International Journal of Space-Based and Situated Computing 5 (1) (2015) 39–52.

[40] S. Gächter, B. Herrmann, C. Thöni, Trust, voluntary cooperation, and socio-economic background: survey and experimental evidence, J. of Economic Behavior & Organization 55 (4) (2004) 505–531.

[41] T. Grandison, M. Sloman, A survey of trust in internet applications, IEEE Communications Surveys & Tutorials 3 (4) (2000) 2–16.

[42] A. Jøsang, R. Ismail, C. Boyd, A survey of trust and reputation systems for online service provision, Decision support systems 43 (2) (2007) 618–644.

[43] M. Momani, S. Challa, Survey of trust models in different network domains, arXiv preprint arXiv:1010.0168.

[44] W. Sherchan, S. Nepal, C. Paris, A survey of trust in social networks, ACM Computing Surveys (CSUR) 45 (4) (2013) 47.

[45] P. R. Vamsi, K. Kant, Systematic design of trust management systems for wireless sensor networks: A review, in: Advanced Computing & Communication Technologies (ACCT), 2014 Fourth International Conference on, IEEE, 2014, pp. 208–215.

[46] G. Fortino, P. Trunfio, Internet of things based on smart objects: Technology, middleware and applications, Springer, 2014.

[47] J. Heidemann, M. Klier, F. Probst, Online social networks: A survey of a global phenomenon, Computer Networks 56 (18) (2012) 3866–3878.

[48] J. Zhan, X. Fang, Social computing: the state of the art, International Journal of Social Computing and Cyber-Physical Systems 1 (1) (2011) 1–12.

[49] A. Aikebaier, T. Enokido, M. Takizawa, Trustworthy group making algorithm in distributed systems, Human-centric Computing and Information Sciences 1 (1) (2011) 6. doi:10.1186/2192-1962-1-6.

[50] T. Huynh, N. Jennings, N. Shadbolt, An integrated trust and reputation model for open multi-agent systems, Autonomous Agents and Multi-Agent Systems 13 (2) (2006) 119–154.

[51] C. Dellarocas, Designing reputation systems for the social web, SSRN Electronic Journal.

[52] Y. Kim, H. Song, Strategies for predicting local trust based on trust propagation in social networks, Knowledge-Based Systems 24 (8) (2011) 1360–1371.

[53] P. Massa, P. Avesani, Trust metrics on controversial users: Balancing between tyranny of the majority, International Journal on Semantic Web and Information Systems (IJSWIS) 3 (1) (2007) 39–64.

[54] C. Ziegler, G. Lausen, Spreading activation models for trust propagation, in: e-Technology, e-Commerce and e-Service, 2004. EEE'04. 2004 IEEE International Conference on, IEEE, 2004, pp. 83–97.

[55] J. Golbeck, Computing and applying trust in web-based social networks, in: PhD Thesis, University of Maryland, Department of Computer Science, 2005.

[56] J. Golbeck, J. Hendler, Inferring binary trust relationships in web-based social networks, ACM Transactions on Internet Technology 6 (4) (2006) 497–529.

[57] R. Guha, R. Kumar, P. Raghavan, A. Tomkins, Propagation of trust and distrust, in: Proc. of the 13th International Conference on World Wide Web, ACM, 2004, pp. 403–412.

[58] P. Massa, P. Avesani, Trust-aware recommender systems, in: Proc. of the 2007 ACM Conference on Recommender systems, ACM, 2007, pp. 17–24.

[59] M. Jamali, M. Ester, Trustwalker: a random walk model for combining trust-based and item-based recommendation, in: Proc. of the 15th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, ACM, 2009, pp. 397–406.

[60] S. J. Brams, P. C. Fishburn, Voting procedures, Handbook of social choice and welfare 1 (2002) 173–236.

[61] N. R. Council, et al., Public participation in environmental assessment and decision making, National Academies Press, 2008.

[62] L. Xia, Computational voting theory: game-theoretic and combinatorial aspects, Ph.D. thesis, Duke University (2011).

[63] T. C. Beierle, J. Cayford, Democracy in practice: Public participation in environmental decisions, Resources for the Future, 2002.

[64] Y. Chevaleyre, U. Endriss, J. Lang, N. Maudet, A short introduction to computational social choice, SOFSEM 2007: Theory and Practice of Computer Science (2007) 51–69.

[65] V. Conitzer, T. Sandholm, Universal voting protocol tweaks to make manipulation hard, arXiv preprint cs/0307018.

[66] J. Pitt, L. Kamara, M. Sergot, A. Artikis, Formalization of a voting protocol for virtual organizations, in: Proc. of the 4th Int joint Conf. on Autonomous Agents and Multiagent Systems, ACM, 2005, pp. 373–380.

[67] A. Gibbard, Manipulation of voting schemes: a general result, Econometrica: J. of the Econometric Society (1973) 587–601.

[68] M. Satterthwaite, Strategy-proofness and arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions, J. of Economic Theory 10 (2) (1975) 187–217.

[69] M. Levi, A state of trust, Trust and governance 1 (1998) 77–101.

[70] P. Dumouchel, Trust as an action, European J. of Sociology/Archives Européennes de Sociologie 46 (3) (2005) 417–428.

[71] T. Jiang, J. S. Baras, Trust evaluation in anarchy: A case study on autonomous networks., in: INFOCOM, 2006.

[72] S. E. Williams, L. P. Shoo, J. L. Isaac, A. A. Hoffmann, G. Langham, Towards an integrated framework for assessing the vulnerability of species to climate change, PLoS biology 6 (12) (2008) e325.

[73] X.-Y. Xiao, W.-C. Peng, C.-C. Hung, W.-C. Lee, Using sensorranks for in-network detection of faulty readings in wireless sensor networks, in: Proc. of the 6th ACM Int. Work. on Data Engineering for Wireless and Mobile Access, ACM, 2007, pp. 1–8.

[74] J. Caverlee, L. Liu, S. Webb, The socialtrust framework for trusted social information management: Architecture and algorithms, Information Sciences 180 (1) (2010) 95–112.

[75] K. M. Khan, Q. Malluhi, Establishing trust in cloud computing, IT professional 12 (5) (2010) 20–27.

[76] H. Takabi, J. B. Joshi, G.-J. Ahn, Security and privacy challenges in cloud computing environments, IEEE Security & Privacy 8 (6) (2010) 24–31.

[77] S. Sicari, A. Rizzardi, L. A. Grieco, A. Coen-Porisini, Security, privacy and trust in internet of things: The road ahead, Computer networks 76 (2015) 146–164.

[78] C. Stergiou, K. E. Psannis, B.-G. Kim, B. Gupta, Secure integration of iot and cloud computing, Future Generation Computer Systems 78 (2018) 964–975.

[79] F. Bao, I.-R. Chen, Dynamic trust management for internet of things applications, in: Proceedings of the 2012 international workshop on Self-aware internet of things, ACM, 2012, pp. 1–6.

[80] M. Nitti, R. Girau, L. Atzori, A. Iera, G. Morabito, A subjective model for trustworthiness evaluation in the social internet of things, in: Personal Indoor and Mobile Radio Communications (PIMRC), 2012 IEEE 23rd International Symposium on, IEEE, 2012, pp. 18–23.

[81] R. Chen, F. Bao, J. Guo, Trust-based service management for social internet of things systems, IEEE transactions on dependable and secure computing 13 (6) (2016) 684–696.

[82] P. N. Mahalle, P. A. Thakre, N. R. Prasad, R. Prasad, A fuzzy approach to trust based access control in internet of things, in: Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems (VITAE), 2013 3rd International Conference on, IEEE, 2013, pp. 1–5.

[83] S. M. Habib, S. Ries, M. Muhlhauser, Towards a trust management system for cloud computing, in: Trust, Security and Privacy in Computing and Communications (TrustCom), 2011 IEEE 10th International Conference on, IEEE, 2011, pp. 933–939.

[84] S.-K. Chong, J. Abawajy, M. Ahmad, I. R. A. Hamid, Enhancing trust management in cloud environment, Procedia-Social and Behavioral Sciences 129 (2014) 314–321.

[85] L. S. Lai, E. Turban, Groups formation and operations in the web 2.0 environment and social networks, Group Decision and negotiation 17 (5) (2008) 387–402.

[86] M. Chen, S. Mao, Y. Liu, Big data: A survey, Mobile networks and applications 19 (2) (2014) 171–209.