Enhancing the Navigability in a Social Network of Smart Objects: a Shapley-value based Approach

L.Militano[a,*], M.Nitti[b], L.Atzori[b], A.Iera[a]

[a]*University Mediterranea of Reggio Calabria, DIIES Department, Italy*
[b]*University of Cagliari, DIEE Department, Italy*

Abstract

The Internet of Things (IoT) holds the promise to interconnect any possible object capable of providing useful information about the physical world for the benefit of humans' quality of life. The increasing number of heterogeneous objects that the IoT has to manage introduces crucial scalability issues that still need appropriate solutions. In this respect, one promising proposal is the Social IoT (SIoT) paradigm, whose main principle is to enable objects to autonomously establish social links with each other (adhering to rules set by their owners). "Friend" objects exchange data in a distributed manner and this avoids centralized solutions to implement major functions, such as: node discovery, information search, and trustworthiness management. However, the number and types of established friendships affect network navigability.

This issue is the focus of this paper, which proposes an efficient, distributed and dynamic solution for the objects to select the right friends for the benefit of the overall network connectivity. The proposed friendship selection mechanism relies on a game theoretic model and a Shapley-value based algorithm. Two different utility functions are defined and evaluated based on either a *group degree centrality* and an *average local clustering* parameter. The comparison in terms of global navigability is measured in terms of average path length for the interconnection of any couple of nodes in the network. Results show that the group

---

[*]Corresponding author. University Mediterranea of Reggio Calabria, DIIES Department, 89100 Reggio Calabria, Italy. Email: leonardo.militano@unirc.it. Telephone and fax: +39 0965 875276

Email addresses: `leonardo.militano@unirc.it` (L.Militano),
`michele.nitti@diee.unica.it` (M.Nitti), `l.atzori@diee.unica.it` (L.Atzori),
`antonio.iera@unirc.it` (A.Iera)

degree centrality brings to an enhanced degree of navigability thanks to the ability to create a suitable core of hubs.

1. Introduction

With the recent spreading of RFID-tagged objects, smart devices, augmented every-day life objects, and wireless sensor/actuator networks, the distance between physical and virtual worlds is gradually being shortened, leading to the so-called Internet of Things (IoT) paradigm. According to [1], by 2015 the RFID devices alone will reach the number of hundreds of billions; in line with this forecast, Cisco [2] foresees 6.58 connected smart devices per person by 2020. However, the success of IoT applications strongly depends on the implementation of satisfactory solutions to meet key system requirements, such as reliability, scalability, and efficiency. In fact, the large number of heterogeneous and pervasive objects continuously generating sensing data [3] and connecting different realms, ranging from transport to education and from business to home management, offers opportunities to deploy manifold applications and services. At the same time, this calls for effective methodologies for a fast and dynamic discovery of objects offering the cited services. Searching information, data and resources in the IoT emerged as a crucial challenge [4]: in addition to the size of the searching space, sensors are often required to produce data in real-time, which corresponds to highly dynamic readings, as it may happen when tracking the position of an object or sensing humidity/temperature/presence in the surrounding environment. A further complication derives from the shift we are witnessing in the interaction model. From a paradigm where humans look for information provided by objects (human-object interaction) IoT is moving towards a model where objects look for other objects to provide composite services for the benefit of human beings (object-object interaction).

An approach with the potential to properly address the mentioned scalability issues, which is recently gaining a high popularity, is based on the exploitation of social networking notions, as formalized by the Social IoT (SIoT) concept [5]. This is intended as a social network where every network object is capable of establishing social relationships with other things in an autonomous way with respect to the owner but according to rules set by her. Every object can then interact with its friends when needing for some assistance, such as the provisioning of a

2

piece of important information or a key service. This allows for the implementation of distributed and scalable procedures to manage object interactions in the same way as humans do when interacting in the social networks. The resulting process has a distributed nature, as each object searches for its target peer among its friends, which, if unable to directly provide the requested service/information, take further actions by enquiring their friends.

Clearly, the performance of such a kind of process in the SIoT network is strictly subject to the capability of the objects to *replicate* the human *innate* behavior in handling social relationships, e.g., select the right friends, consult the appropriate service provider, evaluate the trustworthiness of the peers and community. To this aim, each object has to store and manage the information relevant to its friends: data about past transactions, quality of service for past interactions, services that can be provided by friends, and so on. An important parameter to consider is the number of relationships that an object establishes, which affects its memory consumption, the use of computational power and battery, and the effectiveness of the service discovery operations. Therefore, the choice of the object to promote as a friend among the potential candidates becomes also a key factor influencing the overall system performance and the computational cost in finding the best set of friends.

The above considerations motivate the research in this paper, whose aim is to define for any object *an efficient strategy to select the right friends in the view of improving the overall network navigability*. An important feature to achieve for the benefit of overall scalability is a friend selection policy which is *distributed* and *dynamic* in its nature. This avoids the need for central controllers to set a-priori rules to establish social ties. Additionally, the navigation of the resulting network structure should be independent from the implemented routing algorithm and from the application triggering the search operations. An initial study in this direction has been presented in [6], where an early analysis assessed the suitability of the game theoretic model to the problem. In this paper, we extend our previous research work by proposing a novel utility function for the objects, which reveals to be a better performing solution for our purposes. Additionally, we present the steps to follow for an approximated computation of the Shapley-value as this is a viable approach considered in the literature on game theory to reduce the reduce the computation complexity and guarantee tractability in real problems. To summarize, the major contributions of the paper are:

- Modeling of the friendship selection process in the SIoT context in terms of a cooperative game, where a Shapley-value based algorithm is proposed to

define of the best set of friends for each object in the network;

- Proposing and comparing two suitable utility functions that carefully model the corresponding game and the preferences in the friendship selection for the objects in the SIoT;

- Presenting an approximated computation of the Shapley-value applied to the specific problem to reduce the computational effort in the management of the relationships;

- Analyzing the performance of the proposed solutions to evaluate the network navigability mainly in terms of the average number of hops for local peer search operations.

The remainder of the paper is organized as it follows. In the next Section, the research background and related works are presented. In Section 3 the friendship selection approach for the SIoT is introduced. In Section 4 and 5 the reference game theoretic notions and the utility functions proposed for the problem are described. In Section 6 an approximated computation of the Shapley-value, which is of utmost importance in the resource-constrained SIoT environment, is given together with a computational complexity analysis. The performance evaluation results are summarized in section 7, while conclusive remarks are given in the last section.

2. Research Background and Related Works

2.1. Social Internet of Things (SIoT)

The idea to use social networking notions within the Internet of Things to allow objects to autonomously establish social relationships is recently gaining fast popularity. The driving motivation is that a social-oriented approach is expected to support the discovery, selection and composition of services and information provided by distributed objects and networks [7], [8], [9] and [10]. In this paper, without losing generality, we refer to the social IoT model proposed in [5] (we use the acronym SIoT to refer to it). According to this model, a set of forms of socialization among objects exist. The *parental object relationship* is defined among similar objects built in the same period by the same manufacturer where the role of family is played by the production batch. The objects can establish *co-location object relationship* and *co-work object relationship*, like humans do when they share personal (e.g., cohabitation) or public (e.g., work) experiences.

A further type of relationship is defined for objects owned by the same user (mobile phones, game consoles, etc.) which is called as *ownership object relationship*. This latter relationship is established when objects come into contact, sporadically or continuously, for reasons purely related to relations among their owners (e.g., devices/sensors belonging to friends) and it is named *social object relationship*.

All of the above mentioned relationships within the SIoT platform are created and updated on the basis of the objects features (such as: object type, computational power, mobility capabilities, brand) and activity (frequency in meeting the other objects). The parental and ownership relationships are determined by the static characteristics of the object (or slowly varying characteristics): type, brand, ownership. The other kinds of relationship are determined by the movement of the object and by the other objects it comes across. To manage the resulting network and relationships, the SIoT architecture foresees four major components [5]. The *relationship management* introduces into the SIoT the intelligence that allows objects to start, update, and terminate relationships. This is implemented in the Cloud, in the object gateways, and in the objects themselves when capable of implementing the relevant logic. Clearly, the configuration of these functions is controlled by the object owner; accordingly, the resulting links are asymmetrical. The second component in the SIoT architecture is the *service discovery* that has the purpose to find which objects can provide the required service in the same way humans seek for friendships and information. The third component is the *service composition*, which enables the interaction among objects, and the fourth component is the *trustworthiness management* [10] which aims at understanding how the information provided by other members has to be processed.

In [11], an implementation of a SIoT architecture is described (and its open source version is available at the URL http://www.social-iot.org). The platform introduces a central server which implements the functions needed to register an object, to configure information about the objects, to enable the users to specify the object's behavior and to create and manage the relationships of every object. The server pushes these information to the objects when needed, for example to activate the discovery of a particular service. This solution allows even an object with limited computation capabilities to be able to create and manage its own relationships. This implementation is taken as a reference in this paper as it transfers the burden of handling an indefinite number of relationships in the objects to the server. In particular, we foresee that the server will actually compute the Shapley-value for the right friendship selection and push the final information to the objects. Regardless of the possible implementation, objects with low computation capabilities can benefit from the support of other nodes, friends or dedicated

gateway, to complete demanding tasks, following the Fog Computing paradigm [12].

2.2. Node and Service Discovery in IoT

Searching for objects, data and services in the IoT is a crucial challenge especially in real-time environments [4]. Several approaches for real-time search have been proposed in the literature, but none of them is still offering a complete and satisfactory solution. For instance Snoogle/Microsearch [13] [14] and MAX [15] only perform local searches without taking into account the global domain; Global Sensor Networks (GSN) [16] supports searches on static metadata, whereas Dyser [17] considers only keywords as a query language and does not consider the object contexts. Moreover, a common feature of all these search engines is that they are based on a centralized architecture and, as such, cannot scale properly with the expected rapidly increasing number of devices and the relevant number of queries. Node discovery through social tools has been proposed in [18] and in [19] where node discovery and global resource discovery protocols for the IoT are proposed. In particular, a resolution infrastructure called *digcovery* is defined for maximizing efficiency and sustainability of deployments.

The belief that objects would be able to navigate the SIoT network *with only local information* is founded on the works of the sociologist Stanley Milgram [20] and the computer scientist Jon Kleinberg [21]. Milgram studied the small-world phenomenon and demonstrated that people are tied by short chains of acquaintances, whereas Kleinberg concluded that there are structural clues in a social network that help people to efficiently find a short path even without a global knowledge of the network. Simple proposals to address these issues have been recently introduced, but the followed strategies are simple and the performance has only been analyzed in terms of global [22] or local [23] network navigability.

2.3. Game Theory in Networking

Game theory is an analytical framework that attempts to analyze the behavior of rational entities with their own interests in reciprocal interactions [24]. Starting from the economic field [25], during the last decades game theory has found successful applications to several other areas. A large number of contributions in the literature can be found, dealing with models for wired and wireless communications [26]. Just to mention some fields of application we can list radio resource management [27], cooperation in wireless ad hoc networks [28], pricing schemes in cognitive wireless networks [29] or in heterogeneous wireless networks [30]. Among cooperative game based contributions, the so-called *coalitional games*

[31] are applied in several fields, such as to study fairness and cooperation gains in virtual MIMO systems in [32], packet forwarding issues in ad-hoc networks in [33], task allocation problems in a software system in [34], whereas in [35] and [36] it is applied to introduce a fair energy consumption cost-distribution in a wireless cooperative cluster.

In value or cost-sharing game theoretic applications, one of the most used solution concepts is the Shapley-value [37], [38], [39]. Thanks to its intrinsic capability to capture the contribution of the single players to different coalitions of players, the Shapley-value has found several applications both in networking and social analysis. For instance in [40] it is adopted for a monetary cost analysis for a fair content sharing by both network/service providers and the end-users. An interesting application of the Shapley-value is in the domain of social networks and networks in general, where it is adopted as a measure of the relative importance of the single nodes. For instance, in [41] the authors propose a solution to understand which individual is more important than others in common problems like scientists who collaborate in published articles, or employees of a company who participate in projects. Closer to the contribution of this paper, the Shapley-value has been applied to social networking problems in [42], [43] and [44]. In particular, [42] a set of new *centrality* measures are proposed based on cooperative game theoretic notions. An analysis of the Shapley-value for network centrality has been presented also in [43], with results demonstrating the opportunities for efficiency gains. In [44] the focus is on the information diffusion problem in social networks with particular focus on the target set selection issue. The proposed solution is to select a subset of influential players in a social network with a Shapley-value based algorithm. The promising results obtained for the information diffusion problem in social networking suggest that the Shapley-value has interesting features so that it can also be applied for the friendship selection problem in the SIoT studied in this paper.

3. The Proposed Friendship Selection in the SIoT

The problem of how to select an effective set of friendship relationships among the possible candidates has been addressed in the past in the context of information diffusion in social networking problems. Usually, a score is assigned to each member of the set of potential devices [43], where the selected score somehow corresponds to the importance of that device for the application at hand. However, such a conventional approach suffers from the main intrinsic limitation that it only considers the relative importance of objects as stand-alone entities. Contrar-

ily, a key requirement in the SIoT is to understand the importance of each object in terms of its contribution to a certain *utility* when combined with other nodes [45], [46]. The flexibility, which comes from the ability to take into account the contributions of all possible combinations of objects (rather than considering just one node at a time), is lacking in conventional centrality measures. This is a crucial limitation in many applications and represents the reason why game theoretic network centrality measures have been proposed in research activities relevant to social networks of humans. The promising results obtained in this field, suggest that the Shapley-value has interesting features that can also be exploited for the friendship selection problem in the SIoT studied in this paper. The approach we propose maps the friendship selection process in the SIoT onto the coalition formation problem in a corresponding cooperative game. The Shapley-value of the objects in the game represents the importance of an object and is used to set the friendship preferences.

Before going into the details of the game theoretic modeling, we give details on the friendship selection algorithm proposed for the SIoT. For simplicity in the analysis, we assume that the objects in the SIoT have a commonly shared criterion to decide whether to consider another object as a candidate object for a friendship relation; this criterion is *symmetric*. For instance, if an object i meets the given criterion with another object j, then also node j meets the criterion w.r.t. node i. As described in Section 2.1, a friendship request is triggered when some conditions are satisfied and these conditions depend on the friendship type; e.g. new OORs will be formed when a user registers a new object in the SIoT, while a SOR will be requested when two objects have met for a certain number of times.

We assume that each object can establish up to a maximum number of friendships N_{max} according to its resources. When two nodes met the criterion to establish a new friendship, then a new friendship is directly established if the list of friends for the two objects is less than N_{max}. If this is not the case, then a friendship selection algorithm is triggered to choose the most influential N_{max} friends among the candidate objects. This procedure is reported in Algorithm 1.

The proposed friendship selection algorithm aims at selecting the objects that offer the *maximum marginal contribution* to the global services in the SIoT. To this aim, the candidate friend and the old ones are considered at the same level. Accordingly, a marginal contributions for each one is computed by making use of a cooperative coalitional game modeling, as it will be described in the remainder of the paper. The set of candidate objects are the players in the cooperative game and the Shapley-value is used to compute the marginal contributions of the individual players to the overall value achieved. At the beginning, based on the

Shapley-value, a ranking list of the candidate objects is computed, and then the top N_{max} objects are selected. If nodes i and j are mutually in the top N_{max} objects of the respectively computed ranking lists, then the friendship is established. In some cases the establishment of a new friendship may require an old one to be removed. When this happens, the choice of the friendship to be removed is again driven by the Shapley-value based ranking list, with the only constraint that a node cannot refuse or discard relationships if this action is going to isolate a node.

Algorithm 1: Proposed friendship selection algorithm

---

1   A new candidate friend y is encountered
2   **if** N_{max} *is reached* **then**
3   Compute marginal contribution for existing friends and y
4   Rank the nodes in a list L in a descending order
5   **if** *y is among the first N_{max} nodes* **then**
6   Send a friendship request to y
7   **if** *the request is accepted* **then**
8   Create a new friendship
9   Terminate the relationship with the node in the lowest position of L
10   **end**
11   **else**
12   Discard node y
13   **end**
14 **else**
15   Send a friendship request to y
16   **if** *the request is accepted* **then**
17   Create a new friendship
18   **end**
19 **end**

---

4. Game-Theoretic Notions for Network Navigability

The process of assigning a value to each node/object in a network can be naturally modeled as a coalitional game $G = <\mathcal{N}, v(\cdot)>$ with transferable utilities (TU), where $\mathcal{N}$ is the set of N players and $v(\cdot)$ is a value function. A value function $v(\mathcal{S})$ is a mapping from a nonempty coalition $\mathcal{S} \subseteq \mathcal{N}$ to a real number. Thus, given a subset $\mathcal{S}$ of $\mathcal{N}$, we call $v(\mathcal{S})$ the value of the coalition $\mathcal{S}$. It represents the maximum aggregated payoff available for division among players who are members of $\mathcal{S}$ when working together without the help of players in $\mathcal{N}\backslash\mathcal{S}$. The set of players $\mathcal{N}$ is called the *grand coalition* and $v(\mathcal{N})$ is called the value of the grand coalition.

Given a value definition for a coalition, the challenge is to allocate the value among the players forming the coalition. In other words, finding a solution for the game means to find a vector $x \in \mathfrak{R}^N$ that represents the value allocation to each player in the coalition. In particular, an allocation is a vector $x = (x_1, \cdots, x_n)$ where x_i denotes the value associated with player $i \in \mathcal{N}$. For any coalition $\mathcal{S} \subseteq \mathcal{N}$ we denote by $x(\mathcal{S})$ the value $\sum_{i \in \mathcal{S}} x_i$. An allocation $x = (x_1, \cdots, x_n)$ is said to be a *pre-imputation* if it satisfies the *efficiency* property and it is said to be an *imputation* if it satisfies both the *efficiency* and *individual rationality* properties. In particular, an allocation $x = (x_1, \cdots, x_n)$ satisfies the *efficiency* property if $x(\mathcal{N}) = v(\mathcal{N})$ and the allocation is a *feasible payoff profile* (or pre-imputation) of G. Then, if the allocation is such that $x_i \geq v(\{i\})$ (the allocated value is greater or equal than the value for the player staying alone in singleton coalition $\{i\}$) for each $i \in \mathcal{N}$ then the *individual rationality* is satisfied.

The set of all imputations of the game G is denoted by $X(G)$. An *outcome* for G is an imputation from $X(G)$ that specifies the distribution of the value to any player of the game. A typical requirement of a good outcome is to be "stable" with respect to the possibility that subsets of players find convenient to deviate from it and to form alternative coalitions. The set of such stable outcomes is known as the *core* of the game.

Definition 1 (Core [47]). *The* core $C(G)$ *of a TU game* $G = < \mathcal{N}, v(\cdot) >$ *is the set of all imputations x such that, for each coalition $S \subseteq N$, $x(S) \geq v(S)$.*

We say that if an imputation associated with a coalition is in the core then the coalition is *stable*; otherwise, we say it is *unstable*. Indeed, if $y \notin C(G)$, there exists some coalition S such that $y(S) < v(S)$. Therefore, players in the group S might leave the group N and for the coalitions S obtaining the total value $v(S)$, which is greater than what they obtained according to the allocation y.

In general, the core of a game may be empty as well as it may contain an infinite number of imputations. Games with non-empty cores are called *balanced*. An important class of games where the core is always non-empty is the class of *convex* games. A game is said *convex* if, for every pair of coalitions S and T, $v(S \cup T) + v(S \cap T) \geq v(S) + v(T)$. It can be shown that this holds if the value function is *supermodular*, that is, if $v(S \cup \{i\}) - v(S) \leq v(T \cup \{i\}) - v(T)$, for each pair of coalitions $S \subseteq T \subseteq N \setminus \{i\}, \forall \{i\} \in N$.

However, even if the core is not empty, it remains the problem of choosing an outcome out of possibly infinite many candidates belonging to the core. Thus, solution concepts associated with unique profiles are usually desirable in applications. Among these, the Shapley-value [37] is one of the most used [38], [39].

Indeed, it is an effective approach to the fair allocation of gains obtained from the cooperation among players of a cooperative game [48]. Since some players may contribute more to the total value than others, an important requirement is to fairly distribute gains among the players. To this purpose, in deciding the payoff to be allocated to the players, the Shapley-value accounts for the relative importance of each player to the game.

Definition 2 (Shapley-value [37]). *The Shapley-value of a TU game $G = \langle \mathcal{N}, v(\cdot) \rangle$ is the pre-imputation of G assigning to every player $i \in \mathcal{N}$ the following value*

$$\phi_i(G) = \frac{1}{|\mathcal{N}|!} \sum_{\mathcal{S} \subseteq \mathcal{N} \setminus \{i\}} |\mathcal{S}|!(|\mathcal{N}| - |\mathcal{S}| - 1)![v(\mathcal{S} \cup \{i\}) - v(\mathcal{S})].$$

Accordingly, the Shapley-value assigns a value to each player *i* by taking into account its *average marginal contribution*, where the average is computed over all different sequences of the players so that the grand coalition can be built up from the empty coalition.

It is known that, in any convex game, the Shapley value belongs to the core and thus it is a stable imputation. However, in the general case the Shapley-value may fall outside the core, even if the core is not empty. Thus, the Shapley-value is not necessarily an imputation and it may violate the individual rationality condition.

An alternative equivalent formulation of the Shapley-value for a player *i* is the following:

$$\phi_i(G) = \frac{1}{N!} \sum_{\pi \in \Omega} [v(P(\pi, i) \cup \{i\}) - v(P(\pi, i)] \tag{1}$$

where π is a permutation of the *N* players, Ω is the set of all *N*! permutations of the players, and $P(\pi, i)$ is the set of players preceding player *i* in permutation π.

This solution concept has also a nice axiomatic characterization supporting its notion of fairness, i.e., it is the unique pre-imputation that satisfies the Symmetry, Dummy Player, and Additivity axioms [37]. A natural way to interpret the Shapley-value $\phi_i(G)$ of player *i* is in terms of the average marginal contribution that player *i* makes to any sub-coalition of $\mathcal{N}$ assuming all orderings are equally likely. The Shapley-value takes into account all possible coalitional dynamics and negotiation scenarios among the players and comes up with a single unique way of distributing the value $v(\mathcal{N})$ of the grand coalition among all the players. Thus, the Shapley-value of a player accurately reflects the *bargaining power* of the player and the marginal value it brings to the game.

An additional simplifying assumption usually made when modeling a TU

coalitional game, is that the game is *superadditive*. A game is *superadditive* if the value function of the game is such that for any $\mathcal{S}$ and $\mathcal{T}$ disjoint coalitions ($\mathcal{S} \cap \mathcal{T} = \emptyset$) subsets of $\mathcal{N}$, we have that $v(\mathcal{S} \cup \mathcal{T}) \geq v(\mathcal{S}) + v(\mathcal{T})$. It means that a coalition that is a merger of two or more coalitions will realize a value at least as great as the sum of the values of the coalitions merged. This seems a reasonable assumption for coalitional games and the proposed Shapley-value is usually applied in superadditive games. However, the Shapley-value can also be computed for games with a *non superadditivite* characteristic function [49]. In these cases the Shapley-value will lack the property of *individual rationality*, thus, some players may have values lower than the players they can generate as a singleton coalition. Noteworthy, in this paper the game model does not foresee a value to be shared among players, but it is used to find the most influential objects among a set of candidate friend objects. Therefore, also non superadditive characteristic functions are acceptable for our scope.

5. Utility Functions for the Friendship Selection

A key aspect for the success of the proposed algorithm is the definition of the utility function, which measures the contribution of the candidate objects. By focusing on an object i, which has to determine its N_{max} friendships, in the following we present two possible solutions that meet the system requirements.

5.1. Average local clustering based model

The first utility function we consider for the model is the *average local clustering coefficient* [50]. This is defined as: $\overline{C} = \frac{1}{n} \sum_{j=1}^{n} Clocal_j$ where n is the number of players in a coalition and $Clocal_j$ is the *local clustering coefficient* defined as: $Clocal_j = \frac{2 \cdot e_j}{k_j \cdot (k_j - 1)}$, where k_j is the number of neighbors for node j and e_j is the number of edges among them.

The TU (Transferable Utility)-game $G = \langle \mathcal{N}, v(\cdot) \rangle$ is modeled on the set of $N_{max} + 1$ candidate friends for node i plus the node i itself, thus we have that $N = N_{max} + 2$. This number derives from the consideration that the selection algorithm is triggered only when a new object meets a given criterion to become a friend of object i and this additional object would make the number of friends go beyond the threshold N_{max}. Moreover, also node i must be considered, as the utility function is directly influenced by the presence of node i in any of the coalitions of

objects. In details, we define the value function as:

$$v(\mathcal{S}) = \left(1 - \frac{1}{|\mathcal{S}|} \sum_{i \in \mathcal{S}} Clocal_i\right) \; \forall \mathcal{S} \subseteq \mathcal{N}, \; \text{ and } v(\emptyset) = 0. \tag{2}$$

The driving motivation for this utility function to be considered comes from Kleinberg's findings [21]. In particular, a high value of the average local cluster coefficient allows the nodes to quickly reach the nodes with many connections in the network. However, if the nodes are connected to friends with high value of C_{local}, then clusters of friends are created in the network and, in the extreme case, they can form an isolated subnet, so that the navigability of the network is highly limited. Navigability is thus assured by the ability to reach nodes with low values of local clustering, i.e. by adopting the complementary value, since they are hardly reachable through other paths. This gives us the possibility to perform *local choices* that will guarantee the *global navigability* of the network to be kept at acceptable levels.

Noteworthy, this utility function is *non superadditive* as in some topology configurations some nodes, which join a pre-existing set of friends, may actually adversely affect the local clustering coefficient value. A high Shapley-value is assigned to players giving a high average marginal contribution to all possible permutations of coalitions among the players. This means that a player has a high Shapley-value when it gives a positive contribution to the utility function. In particular, we can identify two sample cases for topologies where a positive or a negative Shapley-value may occur for a node. In particular we may refer to a fully meshed network and a star topology network..... **TBC**. Based on the value function defined in equation 2, we compute the Shapley-value for all the objects in the game and sort the players in decreasing order of their Shapley-value. From the so-constructed rank list, the candidate objects are selected one at the time by scrolling the list in a top-down order (clearly, node i itself will not be considered as a candidate friend). If the friendship with an object is already active, then nothing happens and the subsequent node in the list is selected until N_{max} nodes are selected from the list. If the selected object has not been considered as a friend yet, then the node tries to form a new friendship relationship. If the new friendship request is accepted, then an old friendship must be closed to meet the constraint on the N_{max} number of friendships of node i. Also for this choice the Shapley-based rank list is used, and the less influential node is selected as the one to be removed.

To better understand the behavior of the proposed solution, let us consider the sample study case shown in Figure 1. In particular, objects 1 and 5 meet
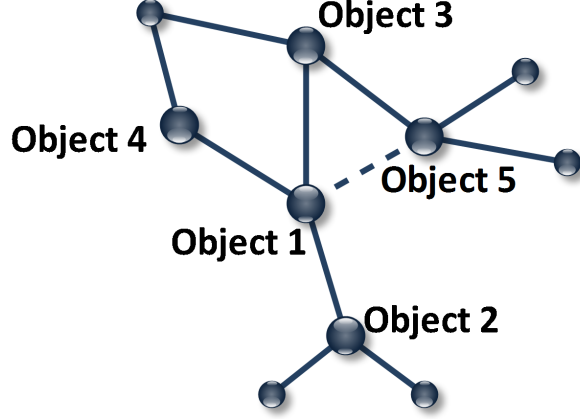
Figure 1: Friendship selection sample scenario.

the criterion for a new potential friendship (see dashed line in the Figure) and the proposed algorithm is triggered to possibly update the friendships for the involved objects. As an example, let us focus the attention on object 1 and the computation of its list of preferences for the friendships (a similar analysis can be repeated for object 5) in the case where the maximum number of friends per object 1 is set to $N_{max} = 3$. This means that the three objects with the highest Shapley-value shall be selected as its friends. In the scenario of the figure we can model the game as follows: $\mathcal{N} = \{1, 2, 3, 4, 5\}$; $v(1) = v(5) = 1 - 0.167 = 0.833$; $v(2) = v(4) = v(24) = 1$; $v(3) = 1 - 0.333 = 0.667$; $v(12) = v(14) = v(25) = v(45) = 1 - 1/2 \cdot (0.167 + 0) = 0.917$; $v(13) = v(35) = 1 - 1/2 \cdot (0.167 + 0.333) = 0.75$; $v(15) = v(23) = v(34) = 1 - 1/2 \cdot (0.167 + 0.167) = 1 - 1/2 \cdot (0 + 0.333) = 0.833$; $v(123) = v(134) = v(235) = v(345) = 1 - 1/3 \cdot (0.167 + 0 + 0.333) = 0.833$; $v(124) = v(245) = 1 - 1/3 \cdot (0 + 0 + 0.167) = 0.944$; $v(125) = v(145) = v(234) = 1 - 1/3 \cdot (0.167 + 0 + 0.167) = 1 - 1/3 \cdot (0 + 0.333 + 0) = 0.889$; $v(135) = 1 - 1/3 \cdot (0.167 + 0.333 + 0.167) = 0.778$; $v(1234) = v(2345) = 1 - 1/4 \cdot (0.167 + 0 + 0.333 + 0) = 0.875$; $v(1235) = v(1345) = 1 - 1/4 \cdot (0.167 + 0 + 0.333 + 0.167) = 0.833$; $v(1245) = 1 - 1/4 \cdot (0.167 + 0 + 0 + 0.167) = 0.917$; $v(12345) = 1 - 1/5 \cdot (0.167 + 0 + 0.333 + 0 + 0.167) = 0.867$.

By computing the marginal contributions of the players based on the Shapley-value we obtain the results in Table 1 (note that not only a subset of the 120 permutations are listed due to length constraints). We observe that the order of

14

preferred friendships for object 1 are respectively, object 2, object 4 and object 5. Since $N_{max} = 3$, object 1 will try to form a new friendship with object 5, since it is already friend with object 2 and 4, and consider object 3 only if any of the preferred friendships is not accepted by the inquired object otherwise it will terminate this friendship.

Table 1: Shapley-value for sample scenario in Figure 1 and average local clustering as utility function.

Case	Object 2	Object 3	Object 4	Object 5
12345	v(12)-v(1)=0.084	v(123)-v(12)=-0.083	v(1234)-v(123)=0.042	v($\mathcal{N}$)-v(1234)=-0.008
12354	v(12)-v(1)=0.084	v(123)-v(12)=-0.083	v($\mathcal{N}$)-v(1235)=0.034	v(1235)-v(123)=0
12435	v(12)-v(1)=0.084	v(1234)-v(124)=-0.069	v(124)-v(12)=0.028	v($\mathcal{N}$)-v(1234)=-0.008
12453	v(12)-v(1)=0.084	v($\mathcal{N}$)-v(1245)=-0.05	v(124)-v(12)=0.028	v(1245)-v(124)=-0.027
12534	v(12)-v(1)=0.084	v(1235)-v(125)=-0.056	v($\mathcal{N}$)-v(1235)=0.034	v(125)-v(12)=-0.028
		$\cdots$		
21345	v(2)-v(∅)=1	v(123)-v(12)=-0.083	v(1234)-v(123)=0.042	v($\mathcal{N}$)-v(1234)=-0.008
21354	v(2)-v(∅)=1	v(123)-v(12)=-0.083	v($\mathcal{N}$)-v(1235)=0.034	v(1235)-v(123)=0
21435	v(2)-v(∅)=1	v(1234)-v(124)=-0.069	v(124)-v(12)=0.028	v($\mathcal{N}$)-v(1234)=-0.008
21453	v(2)-v(∅)=1	v($\mathcal{N}$)-v(1245)=-0.05	v(124)-v(12)=0.028	v(1245)-v(124)=-0.027
21534	v(2)-v(∅)=1	v(1235)-v(125)=-0.056	v($\mathcal{N}$)-v(1235)=0.034	v(125)-v(12)=-0.028
		$\cdots$		
31245	v(123)-v(13)=0.083	v(3)-v(∅)=0.667	v(1234)-v(123)=0.042	v($\mathcal{N}$)-v(1234)=-0.008
31254	v(123)-v(13)=0.083	v(3)-v(∅)=0.667	v($\mathcal{N}$)-v(1235)=0.034	v(1235)-v(123)=0
31425	v(1234)-v(134)=0.042	v(3)-v(∅)=0.667	v(134)-v(13)=0.083	v($\mathcal{N}$)-v(1234)=-0.008
31452	v($\mathcal{N}$)-v(1345)=0.033	v(3)-v(∅)=0.667	v(134)-v(13)=0.083	v(1345)-v(134)=0
31524	v(1235)-v(135)=0.056	v(3)-v(∅)=0.667	v($\mathcal{N}$)-v(1235)=0.034	v(135)-v(13)=0.028
		$\cdots$		
41235	v(124)-v(14)=0.028	v(1234)-v(124)=-0.069	v(4)-v(∅)=1	v($\mathcal{N}$)-v(1234)=-0.008
41253	v(124)-v(14)=0.028	v($\mathcal{N}$)-v(1245)=-0.05	v(4)-v(∅)=1	v(1245)-v(124)=-0.027
41325	v(1234)-v(134)=0.042	v(134)-v(14)=-0.083	v(4)-v(∅)=1	v($\mathcal{N}$)-v(1234)=-0.008
41352	v($\mathcal{N}$)-v(1345)=0.033	v(134)-v(14)=-0.083	v(4)-v(∅)=1	v(1345)-v(134)=0
41523	v(1245)-v(145)=0.028	v($\mathcal{N}$)-v(1245)=-0.05	v(4)-v(∅)=1	v(145)-v(14)=-0.028
		$\cdots$		
51234	v(125)-v(15)=0.056	v(1235)-v(125)=-0.056	v($\mathcal{N}$)-v(1235)=0.034	v(5)-v(∅)=0.833
51243	v(125)-v(15)=0.056	v($\mathcal{N}$)-v(1245)=-0.05	v(1245)-v(125)=0.028	v(5)-v(∅)=0.833
51324	v(1235)-v(135)=0.056	v(135)-v(15)=-0.056	v($\mathcal{N}$)-v(1235)=0.034	v(5)-v(∅)=0.833
51342	v($\mathcal{N}$)-v(1345)=0.033	v(135)-v(15)=-0.056	v(1345)-v(135)=0.056	v(5)-v(∅)=0.833
51423	v(1245)-v(145)=0.028	v($\mathcal{N}$)-v(1245)=-0.05	v(145)-v(15)=0.056	v(5)-v(∅)=0.833
ϕ_i	**0.243**	**0.069**	**0.243**	**0.156**

5.2. Group degree centrality based model

The second utility function we consider is the degree centrality [51] of the objects in the considered coalition, i.e. the union of all the distinct friends of the objects in the coalition. Also in this case, the TU-game $G = \langle \mathcal{N}, v(\cdot) \rangle$ is modeled by considering $N = N_{max} + 2$ as given by the set of candidate friends plus the node i itself. In details, we define the value function as:

$$v(S) = \bigcup_{i \in S} F_i \qquad \forall S \subseteq N, \text{ and } v(\emptyset) = 0$$

where F_i is the set of friends of node i in coalition S. Noteworthy, this utility function is readily *superadditive* as having larger sets of friends will in no case reduce the number of objects that can be reached. Based on the value function defined above, the Shapley-value is then computed for the players (also in this case node i itself is not considered) and their values are sorted in a decreasing order to select the N_{max} most influential objects according to the procedure explained in the previous section.

Let us consider again the sample scenario in Figure 1 to compute Shapley-value according to the group degree centrality as utility function. Let the attention be again on object 1 and consider its list of preferences for the friendships with $N_{max} = 3$. With reference to the plotted scenario, we can model the game as follows: $N = \{1, 2, 3, 4, 5\}; v(1) = v(5) = v(34) = 5; v(2) = v(3) = 4; v(4) = 3; v(12) = v(15) = v(23) = v(45) = v(345) = 7; v(13) = v(14) = v(24) = v(35) = v(134) = 6; v(25) = v(123) = v(124) = v(135) = v(145) = v(234) = v(1234) = v(1345) = 8; v(125) = v(235) = 9; v(245) = v(1235) = v(1245) = v(2345) = v(N) = 10$. By computing the marginal contributions of the players based on the Shapley-value we obtain the results in Table 2 (note that not only a subset of the 120 permutations are listed due to length constraints). Observe that the order of preferred friendships for object 1 are respectively, object 5, object 2 and object 3. Since $N_{max} = 3$, object 1 will try to form a friendship with object 5, since it is already friend with object 2 and 3, and consider object 4 only if any of the preferred friendships is not accepted by the inquired object otherwise it will terminate this friendship.

6. An Approximated Computation for the Shapley-value

The Shapley-value is known to be a computationally heavy solution as all the $N!$ permutations of the players have to be considered to evaluate the average marginal contribution of a player and thus, its importance. In particular, the computational complexity of the Shapley-value computation in (1) is $O\left(\left(\frac{n}{e}\right)^n\right)$, where n is the number of objects involved in the friendship selection process (not the whole network). This computational complexity is high, but the platform implementation is based on a central server taking care, among other functions, of the creation and management for the relationships of every object.

Table 2: Shapley-value for sample scenario in Figure 1 and group degree centrality as utility function.

Case	Object 2	Object 3	Object 4	Object 5
12345	v(12)-v(1)=2	v(123)-v(12)=1	v(1234)-v(123)=0	v($\mathcal{N}$)-v(1234)=2
12354	v(12)-v(1)=2	v(123)-v(12)=1	v($\mathcal{N}$)-v(1235)=0	v(1235)-v(123)=2
12435	v(12)-v(1)=2	v(1234)-v(124)=0	v(124)-v(12)=1	v($\mathcal{N}$)-v(1234)=2
12453	v(12)-v(1)=2	v($\mathcal{N}$)-v(1245)=0	v(124)-v(12)=1	v(1245)-v(124)=2
12534	v(12)-v(1)=2	v(1235)-v(125)=1	v($\mathcal{N}$)-v(1235)=0	v(125)-v(12)=2
		...		
21345	v(2)-v($\emptyset$)=4	v(123)-v(12)=1	v(1234)-v(123)=0	v($\mathcal{N}$)-v(1234)=2
21354	v(2)-v($\emptyset$)=4	v(123)-v(12)=1	v($\mathcal{N}$)-v(1235)=0	v(1235)-v(123)=2
21435	v(2)-v($\emptyset$)=4	v(1234)-v(124)=0	v(124)-v(12)=1	v($\mathcal{N}$)-v(1234)=2
21453	v(2)-v($\emptyset$)=4	v($\mathcal{N}$)-v(1245)=0	v(124)-v(12)=1	v(1245)-v(124)=2
21534	v(2)-v($\emptyset$)=4	v(1235)-v(125)=1	v($\mathcal{N}$)-v(1235)=0	v(125)-v(12)=2
		...		
31245	v(123)-v(13)=2	v(3)-v($\emptyset$)=4	v(1234)-v(123)=0	v($\mathcal{N}$)-v(1234)=2
31254	v(123)-v(13)=2	v(3)-v($\emptyset$)=4	v($\mathcal{N}$)-v(1235)=0	v(1235)-v(123)=2
31425	v(1234)-v(134)= 2	v(3)-v($\emptyset$)=4	v(134)-v(13)=0	v($\mathcal{N}$)-v(1234)=2
31452	v($\mathcal{N}$)-v(1345)=2	v(3)-v($\emptyset$)=4	v(134)-v(13)=0	v(1345)-v(134)=2
31524	v(1235)-v(135)=2	v(3)-v($\emptyset$)=4	v($\mathcal{N}$)-v(1235)=0	v(135)-v(13)=2
		...		
41235	v(124)-v(14)=2	v(1234)-v(124)=0	v(4)-v($\emptyset$)=3	v($\mathcal{N}$)-v(1234)=2
41253	v(124)-v(14)=2	v($\mathcal{N}$)-v(1245)=0	v(4)-v($\emptyset$)=3	v(1245)-v(124)=2
41325	v(1234)-v(134)=2	v(134)-v(14)=0	v(4)-v($\emptyset$)=3	v($\mathcal{N}$)-v(1234)=2
41352	v($\mathcal{N}$)-v(1345)=2	v(134)-v(14)=0	v(4)-v($\emptyset$)=3	v(1345)-v(134)=2
41523	v(1245)-v(145)=2	v($\mathcal{N}$)-v(1245)=0	v(4)-v($\emptyset$)=3	v(145)-v(14)=2
		...		
51234	v(125)-v(15)=2	v(1235)-v(125)=1	v($\mathcal{N}$)-v(1235)=0	v(5)-v($\emptyset$)=5
51243	v(125)-v(15)=2	v($\mathcal{N}$)-v(1245)=0	v(1245)-v(125)=1	v(5)-v($\emptyset$)=5
51324	v(1235)-v(135)=2	v(135)-v(15)=1	v($\mathcal{N}$)-v(1235)=0	v(5)-v($\emptyset$)=5
51342	v($\mathcal{N}$)-v(1345)=2	v(135)-v(15)=1	v(1345)-v(135)=0	v(5)-v($\emptyset$)=5
51423	v(1245)-v(145)=2	v($\mathcal{N}$)-v(1245)=0	v(145)-v(15)=1	v(5)-v($\emptyset$)=5
ϕ_i	**2.7**	**1.367**	**1.2**	**2.867**

Still, to reduce the complexity, an alternative approximated solution can be introduced to reduce the computational burden to a polynomial time computation, while not losing the benefits of the proposed solution. The approach followed in this paper is a sampling based approach that works in polynomial time [44]. In particular, instead of considering all possible $N!$ permutations, a randomly sample set $\tilde{\Omega}$ of t permutations is considered so that $t = O(n)$. To this, the following steps are implemented:

1. Randomly generate $t = |\tilde{\Omega}|$ permutations of the players, with π_j a single permutation from the set $\tilde{\Omega}$;
2. for each permutation π_j in $\tilde{\Omega}$ follow the order of nodes in π_j and compute the contribution of each node $i \in \pi_j$ to the utility of the coalition;
3. determine the average contribution of each node to the utility computed on all t permutations in $\tilde{\Omega}$.

The resulting approximated computation of the Shapley-value for player i can be given in this form:

$$\phi_i(G) = \frac{1}{t} \sum_{\pi_j \in \tilde{\Omega}} \left[v(P(\pi_j, i) \cup \{i\}) - v(P(\pi_j, i)) \right] \tag{3}$$

In the following we will discuss how to determine the value for the permutations t to consider. In particular, the price to pay for the polynomial time complexity is the introduction of an error in the exact computation of the preference index, that can be computed based on statistical techniques [44]. Let $\{X_i^1, X_i^2, \cdots, X_i^t\}$ be the random sample of marginal contributions for node i, when considering all the randomly sampled t permutations in $\tilde{\Omega}$. Let $\bar{X}_i = \frac{\sum_{j=1}^{t} X_i^j}{t}$ be an estimator for the Shapley-value for node i. To understand how close the estimator is to the original Shapley-value, we first need to compute the variance of the random variable $\bar{X}_i$, i.e., $\text{Var}(\bar{X}_i) = \sum_{j=1}^{t} \frac{\{X_i^j - \bar{X}_i\}^2}{t(t-1)}$. To evaluate the quality of the estimator, we need to consider the confidence interval so that we have a certain level of confidence that the considered interval contains the real Shapley-value. In particular, if we have that $P(\bar{X}_i - \delta < \phi_i < \bar{X}_i + \delta) = \gamma$, we can say that the probability, or confidence, is γ that the interval $(\bar{X}_i - \delta, \bar{X}_i + \delta)$ will contain the real Shapley-value ϕ_i.

Given these notions, it is possible to construct $(1 - \alpha)100$ percent confidence intervals of the form $\left(\bar{X}_i - z_{\frac{\alpha}{2}} \left[\frac{s}{t^{0.5}} \right], \bar{X}_i + z_{\frac{\alpha}{2}} \left[\frac{s}{t^{0.5}} \right] \right)$. Where s is the standard deviation of the random sample and z is the standard normal random variable. For instance, for $\alpha = 0.05$ we obtain a 95 percent confidence interval. Moreover, the random sample size t is a further parameter to be properly set. From the definition

of confidence interval, we have $(1 − \alpha)100$ percent probability that the estimator $\bar{X}_i$ deviates from the real Shapley-value by less than $z_{\frac{\alpha}{2}} \frac{s}{t^{0.5}}$ which is called as error e. The value for the error e is important as we can find the sample size t required to obtain a $(1 − \alpha)100$ percent confidence interval of width $2e$, $(\bar{X}_i − e, \bar{X}_i + e)$, for the Shapley-value which is: $t \approx \left(\frac{z_{\frac{\alpha}{2}} s}{e} \right)^2$.

For a correct evaluation, the parameters for the approximated computation should be properly set. In particular, we set $\alpha = 0.1$ so that we get a 90 percent confidence interval. The error value e is instead computed as the average value of an initial t' marginal contribution, multiplied by α. For the initial setting for the value of t', we are in line with the observation obtained in [44]. In particular, it is possible to obtain a good approximation for the marginal contributions of the nodes even with a moderate size of t' (i.e., $t' = 60$). Based on this initial set of permutations, we can determine sample size t required to obtain the wished 90% confidence interval of width $2e$ as we defined earlier in this section.
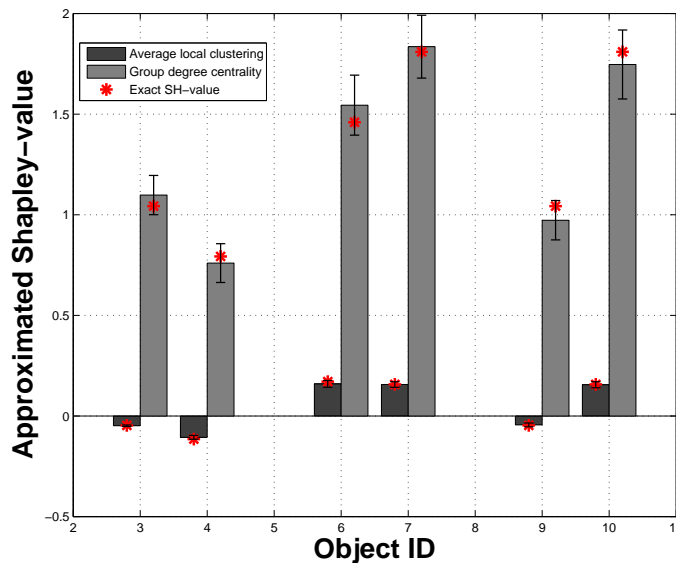

Figure 2: Approximated vs. exact Shapley-value computation in a sample network.

For a clearer view on the effectiveness of the approximated Shapley-value computation, we analyze a sample study case for a small network where $N_{max} = 5$ so that the exact Shapley-value can also be computed. With reference to Figure 2, we present the analysis of the situation where an object has to select its best set of

19

N_{max} friends among six possible choices (i.e., objects with identifier 3, 4, 6, 7, 9, 10). In the plotted results, we report the approximated Shapley-value computation considering the two proposed utility functions (the bars), their confidence interval (note that α was set to 0.1 for a 90 percent confidence interval) and the exact Shapley-value for the objects (marked with a '*'). From the plots, we observe that in some cases the approximated and the exact Shapley-values actually match, whereas in the other cases the exact Shapley-value falls within the confidence interval for the approximated value. However, the most important observation to the scope of our work, is that the object will perform the selection of the same set of friends as for the exact Shapley-value computation. In particular, with the average local clustering utility function the selected friends are objects with ID 7, 10, 6, 3, 9 excluding object with ID 4, whereas for the group degree centrality the preference order for the objects is 6, 10, 7, 3, 9, 4. Noteworthy, even if this is a sample case, the same results yield for any value of N_{max} and network configuration. In fact, the proposed approximated computation is applied to the single objects separately with similar results as those reported in this sample study case.

Considering the proposed approximated Shapley-value computation, the overall computational complexity for the algorithm includes the construction of the preference list (which includes the approximated Shapley-value computation) and the selection of the best nodes from the list. In particular, the marginal contribution for each object is computed in $O(t(n^2))$ for the average local clustering model and in $O(t(n))$ for the group degree centrality model. Then the sorting algorithm to order the list of marginal contributions for each object has a complexity of $O(nlog_2(n))$, whereas the selection of the best nodes from the ordered vector has a constant cost K equal to N_{max}. Thus, the overall computational complexity for the Shapley-based scheme is $O(t(n^2)) + nlog_2(n) + K)$ and $O(t(n)) + nlog_2(n) + K)$ for the average local clustering model and the group degree centrality model respectively, where t is a polynomial in n. This shows how the second utility function introduces a lower computationally burden to the solution.

7. Performance Evaluation

A numerical evaluation has been conducted by using the MATLAB® tool for a wide set of scenarios, to observe the performance of the proposed strategies in terms of network navigability. To perform the simulations we had to generate synthetic data about social networks of objects as no real data is available for significant numbers of nodes. The approach used to construct the synthetic network is based on the following main steps:

1. a social network among humans is analyzed;
2. the social activities of the objects are derived from the previous analysis by considering that objects encounter each other on the basis of the activities of the human owners;
3. characteristics of the resulting network are analyzed;
4. a model is then used to create a synthetic network with properties similar to those observed in the previous step.

We thus rely on the Barabási-Albert model [52], which is able to generate scale-free (independently from the size) networks based on preferential attachment with the same characteristics of desired target social networks. Preferential attachment means that the more connected a node is, the more likely it is to receive new links. The model starts with a small number of nodes and, at each step, it adds a new node with m edges (m is a parameter for the model set to $m = 4$ in our simulations) linked to nodes which are already part of the system. The probability p_i that a new node is connected to an existing node i depends on its degree k_i and on the sum made over all pre-existing nodes j, so that $p_i = k_i/(\sum_j k_j)$ leading to the name preferential attachment. The model is used to create a network of the desired size. In our simulations, we started from the data set of the location-based online social network Brightkite obtained from the Stanford Large Network Dataset Collection [53]. We terminated the generation of the network when 15k nodes and 60k edges have been obtained.

As already discussed in [23], the maximum number of friendship relationships N_{max} for each node is dynamically changed to keep under control the number of hubs in the network. Indeed, a constant value for N_{max} would bring to a flat network without hubs, which would adversely affect the network navigability. Specifically, N_{max} has been increased by a value of 10% whenever

there are x% of N nodes in the network with at least y% of N_{max} friends,

where x represents the maximum percentage of hubs in the network, and y represents the threshold for a node to become a hub.

On the basis of these preliminaries, we proceeded with evaluating the effectiveness of the proposed strategies in terms of network navigability by making use of local routing rules. To this we selected a simple routing rule which works as it follows. Consider a scenario where a given object A wishes to communicate with node B. The first task to perform is to check whether it has a direct connection with object B, that is, B is among its friends. If not, A asks to the friend object

with the *highest connectivity degree*, let say X, to find a route to reach B. Then, object X repeats the same procedure until object B is finally reached.
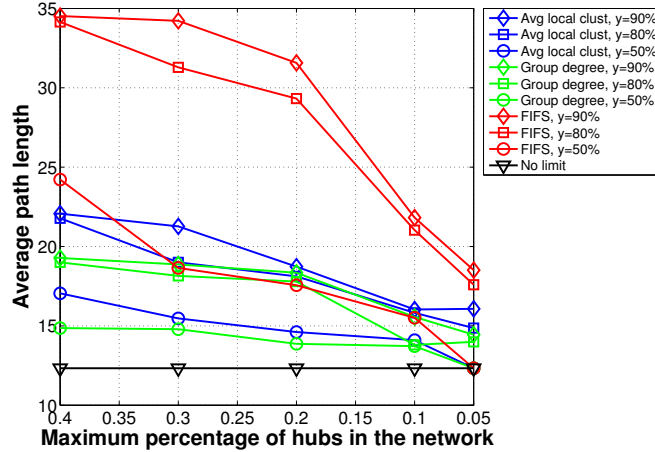

Figure 3: Average path length for different values of the maximum percentage of hubs in the network.

In Figure 3 the average path length is reported, for different values in the range $[0.05 - 0.4]\%$ of the maximum percentage of hubs x in the network and by considering either $y = 50\%$, $y = 80\%$ or $y = 90\%$. The proposed solutions are compared to the case where no limit is set for the number of friends and then no selection strategy is needed (labeled as "no limit" in the plots), and to a *dummy* strategy whereby a node refuses any new request of friendship after reaching N_{max} friends and the connections are static (labeled as "FIFS" in the plots, i.e., First In First Served).

As we can observe from the plots in Figure 3, a reduction in the number of hubs in the network (lower values for x) allows for an improvement in the performance as the average path length is reduced for all the tested solutions, but the "no limit" case. This result is in line with what was suggested by the output of Kleinberg's studies. A similar effect is also obtained when decreasing the threshold for a node to become a hub, i.e., the y-value. In fact, this makes it "easier" to happen that $x\%$ of nodes in the network have at least $y\%$ of N_{max} friends; thus, N_{max} increases more rapidly. However, if we relax too much the control parameters, we fall again in the "no limit" scenario; this is the case of $x = 0.05\%$ and $y = 50\%$. In fact, for this combination of parameters, no matter the strategy adopted, all of them converge to a performance level equal to the "no limit" case. Theoretically,

22

if x is set to $\frac{1}{N}\%$, then for every node reaching a number of friends equal to $y\%$ of N_{max}, the value of N_{max} increases. On the other hand, if we set $y = 0\%$, every node is considered as an hub and N_{max} increases no matter what the value for x is.

When comparing the Shapley-based algorithms to the FIFS strategy, we observe how both utility functions outperform the FIFS solution performing up to 50% better. Additionally, the solution based on the group degree utility performs better than the solution based on the average local clustering coefficient utility. This is true independently from the y threshold being set to 80% or 90%. This result is important as it shows that the ability to reach a larger number of nodes in the network, as the average degree based utility pursues, is more important than the ability to reach nodes with low values of local clustering, as pursued by the second utility function we considered. As was expected, the best performance are always reached with the "not limit" approach, which however has the major drawback that the number of friends managed by each object becomes significantly high, as it is discussed in the following.
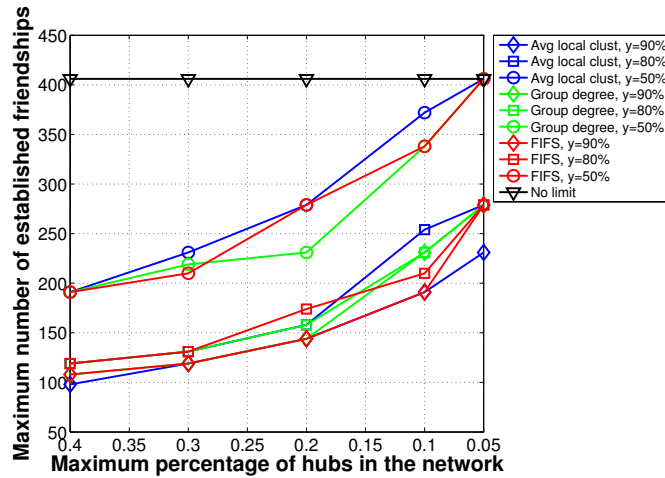

Figure 4: Maximum number of friends for different values of the maximum percentage of hubs in the network.

The second important result we investigated, is the maximum number of friendships established by a node, which is plotted in Figure 4. We can observe how this number is mostly influenced by the value of y rather than the particular strategy implemented (e.g., for $y = 90\%$ the plots are overlapping in many of the tested points). However, it is important to point out how the proposed algorithm is able to drastically decrease the number of friends that a node has to manage w.r.t. to

the "no limit" case. This is an interesting result, as it has a positive effect on the computational burden for the objects.

By comparing Figures 3 and 4, an important observation is that if the number of hubs in the network is low, then each hub has to manage many friends. This feature also means that by using more stringent values for x and y parameters, it is possible to achieve better performance in terms of local navigability, but this goes at the cost of an increase in the memory consumption, computational power, and battery life. To limit this problem, an enhancement for the solution could be to adjust the maximum number of friendships N_{max} based on the node features and then allow only nodes with high computation capabilities, such as vehicles or smart devices, to become hubs.
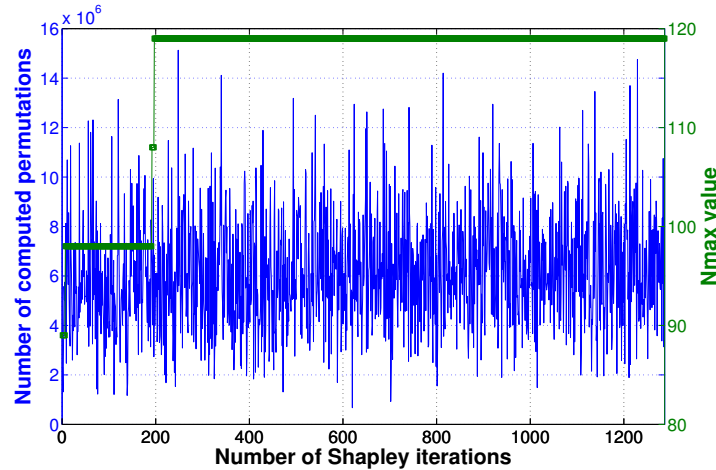

Figure 5: Number of permutations with the corresponding N_{max} value, in the computation of the approximated Shapley-value adopting the average local clustering as utility function. The sample case is presented with $x = 0.4\%$ and $y = 80\%$.

7.1. Influence of the utility function on the approximated Shapley-value computational cost

The computational burden for the approximated Shapley-value is the focus of the next set of plots. In particular, in Figures 5 and 6, we report the number of permutations needed to compute the Shapley-value during the simulations (blue line) and the corresponding value for the maximum number of friends N_{max} (green line) when considering respectively the average local clustering and the group degree centrality as utility function for the algorithm. In particular, a single study case
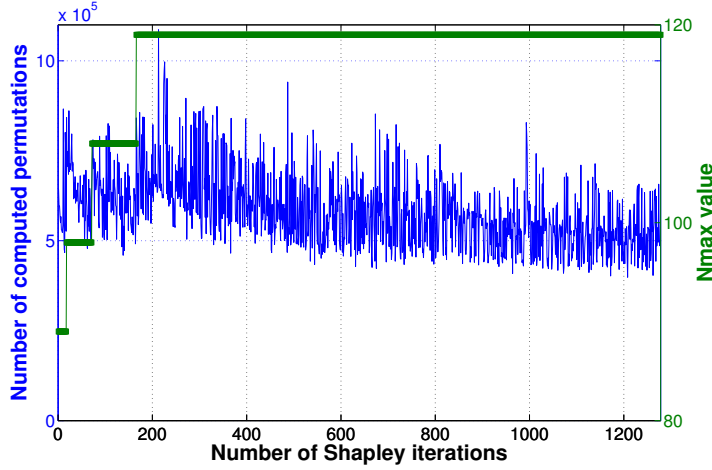
Figure 6: Number of permutations with the corresponding N_{max} value, in the computation of the approximated Shapley-value adopting the group degree centrality as utility function. The sample case is presented with $x = 0.4\%$ and $y = 80\%$

is presented where low restrictive control parameter values are adopted, namely $x = 0.4\%$ and $y = 80\%$, so that more computations are considered. The first observation is that the maximum value reached by the N_{max} parameter is the same in the two cases (this is consistent with the results in Figure 4).

Further interesting observations derive from the careful observation of these plots. First of all, when considering the group degree centrality in Figure 6, we observe that the computational burden (expressed in number of permutations to consider) is influenced by the N_{max} value. In fact, the number of permutations needed to compute the approximated Shapley-value increases every time the maximum number of friends for the nodes is increased. This feature is not observed instead when adopting average local clustering as utility function in Figure 5, where a more constant trend is observed independently from the N_{max} value. This behavior can be justified with reference to the high variance of the utility values in the first t samples computed (please refer to Section 6 for the definition of the t-value). This results in a high number of samples to consider for a correct approximation of the Shapley-value (i.e., the t-value). Differently, in the group degree centrality a change in the value of N_{max} will have a lower impact on the variation of utility. Another observation is related to the number of permutations reduction shown with the group degree centrality model during a single simulation. It is important to remind that the model starts with a small number of nodes and, in consecutive

25

steps, it adds a new node with *m* links to nodes that are already part of the system. According to the approximated Shapley-value computation, when a new friendship request triggers the friendship selection algorithm, the variance of the initial samples of marginal contributions determines the total number of permutations *t* needed to obtain the wished confidence interval (see Section 6 for the details). On the other hand, when a new node joins a pre-existing set of friends, the variance is influenced by the strength of the marginal contributions of the original set of friends. Adopting the group degree centrality model, which forms a set of friends so that as much nodes as possible are reached, the perturbation of a new node joining a pre-existing set of friends is relatively low in terms of the utility function variance. This again leads to a moderate value for the number of samples to consider for a correct approximation of the Shapley-value, the *t*-value. Instead, in the average clustering model a new node can potentially affect the local clustering coefficient value of every other node in the coalition, so that, on average, the variance on the initial set of permutations leads to a high number of total marginal contributions.

8. Conclusion

In this paper we defined the problem of friendship selection in the Social Network of Things, highlighting that if a SIoT is properly created then it can show the characteristics of a small world network and comply with the condition for network navigability. Based on this knowledge, we proposed a model for a distributed friendship selection that relies on the Shapley-value. To this aim, the friendship selection process in the SIoT is mapped onto the coalition formation problem for a corresponding cooperative game. Based on two different utility functions, that meet the constraints for the system, the Shapley-value nicely models the importance of an object in the social IoT network and is thus used to set the friendship preferences. The resulting network navigability, in terms of average number of hops by using local peer search operations, is evaluated and compared with a standard solution where no limit on the friendships is set for the objects. The results showed how a better management of the number of friendships is obtained, at the expense of a negligible increase in the number of hops needed to reach a destination.

A further extension for the implementation of the proposed solution that will be considered in our future research, is to consider enhancing aspects referring to the average path length. In particular, in our simulations we have considered all the possible pairs of nodes to be uniformly distributed over the network. However,

it has been proven that friends share similar interests (bringing to the homophily phenomenon [54]), so that it is highly probable to find another node in the friends list or in the friend of a friend (FOAF) list, thus reducing the average path length among all the pairs of nodes. Node similarity for the routing operations has not been considered so far; indeed, in our simulations nodes try to reach their destination by using only information about the degree of their neighbors. However, external properties could be used to select the right nodes (among available friends) to which the desired service is requested. One of these properties is the profile of the friend involved (accounting for its trustworthiness and the type of relationship that links it to the requester).

References

[1] IDTechEx, R. Das, P. Harrop, RFID forecasts, players and opportunities 2011-2021, IDTechEx, 2011.

[2] D. Evans, The internet of things: how the next evolution of the internet is changing everything, CISCO white paper 1 (2011).

[3] L. Atzori, A. Iera, G. Morabito, The internet of things: A survey, Computer networks 54 (2010) 2787–2805.

[4] D. Zhang, L. Yang, H. Huang, Searching in internet of things: Vision and challenges, in: Parallel and Distributed Processing with Applications (ISPA), 2011 IEEE 9th International Symposium on, 2011, pp. 201–206. doi:10.1109/ISPA.2011.53.

[5] L. Atzori, A. Iera, G. Morabito, M. Nitti, The social internet of things (siot)– when social networks meet the internet of things: Concept, architecture and network characterization, Computer Networks 56 (2012) 3594–3608.

[6] L. Militano, M. Nitti, L. Atzori, A. Iera, Using a distributed shapley-value based approach to ensure navigability in a social network of smart objects, IEEE International Conference on Communications (ICC), London, United Kingdom (2015).

[7] A. Ortiz, D. Hussein, S. Park, S. Han, N. Crespi, The cluster between internet of things and social networks: Review and research challenges, Internet of Things Journal, IEEE 1 (2014) 206–215.

[8] P. Mendes, Social-driven internet of connected objects, in: Proc. of the Interconn. Smart Objects with the Internet Workshop, 2011.

[9] E. A. K. amd N. D. Tselikas, A. C. Boucouvalas, Integrating rfids and smart objects into a unified internet of things architecture, Advances in Internet of Things 1 (2011) 5–12.

[10] M. Nitti, R. Girau, L. Atzori, Trustworthiness management in the social internet of things, Knowledge and Data Engineering, IEEE Transactions on 26 (2014) 1253–1266.

[11] R. Girau, M. Nitti, L. Atzori, Implementation of an experimental platform for the social internet of things, in: Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2013 Seventh International Conference on, IEEE, 2013, pp. 500–505.

[12] F. Bonomi, R. Milito, J. Zhu, S. Addepalli, Fog computing and its role in the internet of things, in: Proceedings of the first edition of the MCC workshop on Mobile cloud computing, ACM, 2012, pp. 13–16.

[13] H. Wang, C. C. Tan, Q. Li, Snoogle: A search engine for pervasive environments, Parallel and Distributed Systems, IEEE Transactions on 21 (2010) 1188–1202.

[14] C. C. Tan, B. Sheng, H. Wang, Q. Li, Microsearch: A search engine for embedded devices used in pervasive computing, ACM Transactions on Embedded Computing Systems (TECS) 9 (2010) 43.

[15] K.-K. Yap, V. Srinivasan, M. Motani, Max: Wide area human-centric search of the physical world, ACM Transactions on Sensor Networks (TOSN) 4 (2008) 26.

[16] K. Aberer, M. Hauswirth, A. Salehi, Infrastructure for data processing in large-scale interconnected sensor networks, in: Mobile Data Management, 2007 International Conference on, IEEE, 2007, pp. 198–205.

[17] K. Romer, B. Ostermaier, F. Mattern, M. Fahrmair, W. Kellerer, Real-time search for real-world entities: A survey, Proceedings of the IEEE 98 (2010) 1887–1902.

[18] A. Jara, P. Lopez, D. Fernandez, J. Castillo, M. Zamora, A. Skarmeta, Mobile digcovery: discovering and interacting with the world through the internet of things, Personal and Ubiquitous Computing 18 (2014) 323–338.

[19] A. J. Jara, P. Lopez, D. Fernandez, J. F. Castillo, M. A. Zamora, A. F. Skarmeta, Mobile digcovery: discovering and interacting with the world through the internet of things, Personal and ubiquitous computing 18 (2014) 323–338.

[20] J. Travers, S. Milgram, An experimental study of the small world problem, Sociometry (1969) 425–443.

[21] J. Kleinberg, The small-world phenomenon: An algorithmic perspective, in: Proceedings of the thirty-second annual ACM symposium on Theory of computing, ACM, 2000, pp. 163–170.

[22] M. Nitti, L. Atzori, I. Cvijikj, Network navigability in the social internet of things, in: Internet of Things (WF-IoT), 2014 IEEE World Forum on, 2014, pp. 405–410. doi:10.1109/WF-IoT.2014.6803200.

[23] M. Nitti, L. Atzori, I. Cvijikj, Friendship selection in the social internet of things: challenges and possible strategies, Internet of Things Journal, IEEE PP (2014) 1–1.

[24] M. Osborne, An introduction to game theory, volume 3, Oxford University Press New York, 2004.

[25] J. Von Neumann, O. Morgenstern, Theory of Games and Economic Behavior (60th Anniversary Commemorative Edition), Princeton university press, 2007.

[26] V. Srivastava, J. O. Neel, A. B. MacKenzie, R. Menon, L. A. DaSilva, J. E. Hicks, J. H. Reed, R. P. Gilles, Using game theory to analyze wireless ad hoc networks., IEEE Communications Surveys and Tutorials 7 (2005) 46–56.

[27] L. Badia, M. Lindström, J. Zander, M. Zorzi, An economic model for the radio resource management in multimedia wireless systems, Computer Communications 27 (2004) 1056–1064.

[28] V. Srinivasan, P. Nuggehalli, C.-F. Chiasserini, R. R. Rao, Cooperation in wireless ad hoc networks, in: INFOCOM 2003. Twenty-Second Annual

Joint Conference of the IEEE Computer and Communications. IEEE Societies, volume 2, IEEE, 2003, pp. 808–817.

[29] D. Niyato, E. Hossain, Z. Han, Dynamic spectrum access in ieee 802.22-based cognitive wireless networks: a game theoretic model for competitive spectrum bidding and pricing, Wireless Communications, IEEE 16 (2009) 16–23.

[30] L. Militano, G. Araniti, Introducing fairness-efficiency trade-off for energy savings in wireless cooperative networks, Wireless Personal Communications 76 (2014) 3–21.

[31] W. Saad, Z. Han, M. Debbah, A. Hjorungnes, T. Basar, Coalitional game theory for communication networks, Signal Processing Magazine, IEEE 26 (2009) 77–97.

[32] S. Mathur, L. Sankar, N. B. Mandayam, Coalitions in cooperative wireless networks, Selected Areas in Communications, IEEE Journal on 26 (2008) 1104–1115.

[33] Z. Han, H. V. Poor, Coalition games with cooperative transmission: a cure for the curse of boundary nodes in selfish packet-forwarding wireless networks, Communications, IEEE Transactions on 57 (2009) 203–213.

[34] O. Shehory, S. Kraus, Methods for task allocation via agent coalition formation, Artificial Intelligence 101 (1998) 165–200.

[35] A. Iera, L. Militano, L. P. Romeo, F. Scarcello, Fair cost allocation in cellular-bluetooth cooperation scenarios, Wireless Communications, IEEE Transactions on 10 (2011) 2566–2576.

[36] L. Militano, A. Iera, A. Molinaro, F. Scarcello, Energy saving analysis in cellular-wlan cooperative scenarios, IEEE Transactions on Vehicular Technology (2014).

[37] L. Shapley, A value for n-person games, Technical Report, DTIC Document, 1952.

[38] S. S. Fatima, M. Wooldridge, N. R. Jennings, An analysis of the shapley value and its uncertainty for the voting game, in: Agent-Mediated Electronic Commerce. Designing Trading Agents and Mechanisms, Springer, 2006, pp. 85–98.

[39] R. Serrano, Cooperative games: Core and shapley value. encyclopedia of complexity and systems science, 2007.

[40] L. Militano, A. Iera, F. Scarcello, A fair cooperative content-sharing service, Computer Networks 57 (2013) 1955–1973.

[41] P. Papapetrou, A. Gionis, H. Mannila, A shapley value approach for influence attribution, in: Machine Learning and Knowledge Discovery in Databases, Springer, 2011, pp. 549–564.

[42] D. Gómez, E. González-Aragüena, C. Manuel, G. Owen, M. del Pozo, J. Tejada, Centrality and power in social networks: a game theoretic approach, Mathematical Social Sciences 46 (2003) 27–54.

[43] K. Aadithya, B. Ravindran, T. Michalak, N. Jennings, Efficient computation of the shapley value for centrality in networks, in: Internet and Network Economics, Springer, 2010, pp. 1–13.

[44] R. Narayanam, Y. Narahari, A shapley value-based approach to discover influential nodes in social networks, IEEE Transactions on Automation Science and Engineering (2010) 1–18.

[45] D. Kempe, J. Kleinberg, É. Tardos, Influential nodes in a diffusion model for social networks, in: Automata, languages and programming, Springer, 2005, pp. 1127–1138.

[46] D. Mocanu, G. Exarchakos, A. Liotta, Node centrality awareness via swarming effects, in: Systems, Man and Cybernetics (SMC), 2014 IEEE International Conference on, 2014, pp. 19–24. doi:10.1109/SMC.2014.6973878.

[47] D. Gillies, Solutions to general non-zero-sum games, volume 40, Contributions to the Theory of Games IV, Annals of Mathematics Studies, H. W. Kuhn and A. W. Tucker editors, 1959.

[48] S. H. Tijs, T. S. Driessen, Game theory and cost allocation problems, Management Science 32 (1986) 1015–1028.

[49] R. Aumann, Some non-superadditive games, and their shapley values, in the talmud, International Journal of Game Theory 39 (2010) 3–10.

[50] D. Watts, S. Strogatz, Collective dynamics of small-worldnetworks, nature 393 (1998) 440–442.

[51] M. G. Everett, S. P. Borgatti, The centrality of groups and classes, The Journal of mathematical sociology 23 (1999) 181–201.

[52] A.-L. Barabási, R. Albert, Emergence of scaling in random networks, science 286 (1999) 509–512.

[53] J. Leskovec, Stanford large network dataset collection, 2015. URL: http://snap.stanford.edu/data/.

[54] H. Bisgin, N. Agarwal, X. Xu, Investigating homophily in online social networks, in: Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on, volume 1, 2010, pp. 533 –536.