Friendship selection in the Social Internet of Things: challenges and possible strategies

Michele Nitti*, Luigi Atzori*, Irena Pletikosa Cvijikj**
*University of Cagliari, Italy, michele.nitti, l.atzori@diee.unica.it
**ETH Zurich, Switzerland, ipletikosa@ethz.ch

***Abstract*—The Internet of Things is expected to be overpopulated by a very large number of objects, with intensive interactions, heterogeneous communications and millions of services. Consequently, scalability issues will arise from the search of the right object that can provide the desired service.**

A new paradigm known as Social Internet of Things (SIoT) has been introduced and proposes the integration of social networking concepts into the Internet of Things. The underneath idea is that every object can look for the desired service using its friendships, in a distributed manner, with only local information.

In the SIoT it is very important to set appropriate rules in the objects to select the right friends as these impact the performance of services developed on top of this social network. In this work we addressed this issue by analyzing possible strategies for the benefit of overall network navigability. We first propose five heuristics which are based on local network properties and that are expected to have an impact on the overall network structure. We then perform extensive experiments, which are intended to analyze the performance in terms of giant components, average degree of connections, local clustering and average path length.

Unexpectedly we discovered that minimizing the local clustering in the network allowed for achieving the best results in terms of average path length. We have conducted further analysis to understand the potential causes, which have been found to be linked to the number of hubs in the network.

***Index Terms*—Internet of Things, social networks, SIoT, navigability, search engine**

I. Introduction

The Internet of Things (IoT) integrates a large number of heterogeneous and pervasive objects that continuously generate information about the physical world. Most of this information is available through standard Web browsers and several platforms already offer application-programming interfaces (APIs) for accessing to sensors and actuators. Accordingly, the IoT technologies make possible to provide new services to end-users in disparate fields, from the environment monitoring to the industrial plants running, from the city management to the house management.

As explained in [1], the search of each specific service provided by the devices in the IoT represents a crucial challenge: the number of objects connected to the network is increasing exponentially, leading to an enormous searching space. According to [2], by 2015 the RFID devices alone will reach hundreds of billions. The network traffic, both in terms of the number of accesses to the devices and of the number of queries received by the search engines, will soon become too large to be managed efficiently by the existing platforms. Additionally, nowadays the interaction model is based on humans looking for information provided by objects (human-object interaction), but in the near-future this model will quickly shift to the object-object interaction, where objects will look for others to provide composite services for the benefit of the humans, increasing the interaction complexity. Consequently, scalability issues will arise from the search of the right object that can provide the desired service.

In this context, several approaches for real-time search have been proposed, such as those described in [3] and [4]. A common feature is that these engines are based on centralized systems and, as such, can not scale properly with the number of devices or/and the number of queries.

To cope with scalability issues of centralized systems, a new paradigm known as Social Internet of Things (SIoT) has been introduced [5]. SIoT proposes the integration of social networking concepts into the IoT solutions. In the SIoT, every node is an object capable of establishing social relationships with other things in an autonomous way according to rules set by the owner.

A SIoT network is based on the idea that every object can look for the desired service by using its relationships, querying its friends and the friends of its friends in a distributed manner, in order to guarantee an efficient and scalable discovery of objects and services following the same principles that characterize the social networks for humans. The assumption that a SIoT network will be navigable is based on the principle of the sociologist Stanley Milgram about the small-world phenomenon. This paradigm refers to the existence of short chains of acquaintances among individual in societies [6]; starting from Milgram's experiment, Kleinberg concluded that there are structural clues that help people to find a short path efficiently even without a global knowledge of a network [7].

According to this paradigm, each object has to store and manage the information related to the friendships, implement the search functions, and eventually employ additional tools such as the trustworthiness relationship module to evaluate the reliability of each friend. Clearly, the number of relationships affects the memory consumption, the use of computational power and battery, and the efficacy of the service search operations. The friendships usefulness varies from friend to friend and then which object to promote as a friend among the potential candidates is the a key aspect for the overall system performance. It results that the selection of the friendships is key for a successful deployment of the SIoT.

Even if social-related measures have been used to exploit the influence of a node, to the best of our knowledge, this is the first time they are used to select a set of nodes in a Social IoT scenario. This issue has been addressed in [8], of which this paper is an extension. Specifically, we have analyzed possible strategies to be implemented by each node when adding new friends taking into account the impact on the network navigability. The major contributions of the paper are the following:

Firstly, we proposed five heuristics which are based on local network properties: neighborhood degree and local clustering. These heuristics are used to rank the nodes in decreasing order and choose the ones that maximize the chosen heuristic.The performance has then been analyzed in terms of global network navigability, i.e., routing is performed by assuming that each object has a view about the global social network topology. From simulations, it resulted that the approach reaching the best results is the one when objects select friends (or substitute old friends) so that on average the resulting friends have a low local neighbor degree.

Secondly, we analyzed how the proposed strategies behave when the routing is performed by each objects only exploiting local information about their friends, namely their degree. In this way each node is not obliged to have the local network topology, reducing the routing complexity. Unexpectedly, it has been discovered that minimizing the local clustering in the network allowed to achieving the best results in terms of average path length, and identified the concentration of hubs as the motivation of this discovery. Accordingly, we proposed a new methodology to dynamically adjust the number of friends allowed per object on the basis of the number of hubs in the network, so that the degree distribution is kept closer to a power law distribution. In this way, we are able to guarantee local network navigability at the limited expenses of the need of a central server monitoring the number of hubs in the network.

The paper is organized as follows. In Section II we present the scenario of the social IoT and provide a quick survey of the solutions for the search of services in the IoT. In Section III we introduce the key aspects of network navigability and the strategies for link selection, whereas Section IV presents the experimental evaluation. In Section V, we show the differences between global and local navigability while Section VI draws the final remarks.

II. Background

A. Social IoT

The idea of using social networking elements in the IoT to allow objects to autonomously establish social relationships is gaining popularity in the last years. The driving motivation is that a social-oriented approach is expected to boost the discovery, selection and composition of services and information provided by distributed objects and networks that have access to the physical world [9], [10], [11] and [12].

Without losing the generality, in this paper 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 are foreseen. The *parental object relationship* (POR) is defined among similar objects, built in the same period by the same manufacturer (the role of family is played by the production batch). Moreover, objects can establish *co-location object relationship* (CLOR) and *co-work object relationship* (CWOR), 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.) that is named *ownership object relationship* (OOR). The last 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); it is named *social object relationship* (SOR). These relationships are created and updated on the basis of the objects' features (such as: type, computational power, mobility capabilities, brand, etc) and activities (frequency in meeting the other objects, mainly).

B. Service search in IoT

In this section, we provide some examples of the existing solutions for service search in IoT context, in order to highlight existing problems. [13] and [4] cope with the large number of real-world entities by using a hierarchy of mediators: the ones in the lower level are responsible for groups of sensors in geographical areas, while the single mediator on the top level maintains an aggregated view of the entire network. These approaches are not scalable in case of frequent data and network changes whereas work well in case of pseudo-static metadata.

In [3], the authors propose a centralized system where objects are contacted based on a prediction model that calculates the probability of matching the query. In this way, the search engine does not need to contact all the sensors leading to good scalability with the number of objects; nevertheless, it is not scalable with the network traffic, since the number of possible results is significantly larger than the number of actual results, so a lot of sensors are contacted for no reason.

III. Reference scenario

A. Distributed search in the IoT

In the same way the search of contents of different kind, such as videos and web pages, is one of the most popular services on the Internet, the search of data from sensors and real-world entities is expected to be a major service in the IoT in the near future. However, the huge number of objects and the frequent changes in their data put a great stress on the service search.

In the SIoT, the objects inherit some capabilities of the humans and mimic their behavior when looking for new friends or services [11]. Indeed, the relationships devised for the SIoT follow the ones studied in sociological and anthropology fields, such as [14] and [15], since the owner sets the rules for their creation. The object then creates and manages several kinds of relationships and uses them to navigate the network, looking for services. The object asks its friends if they can provide a particular service or if they "know", i.e. if they have any connections to, nodes that can provide it.

Figure 1 provides a simple example of a SIoT network, where links represent friendship ties while the bold line is the best route for node 1 to reach the requested service. In this network, when node 1 needs a particular service, it does not send a request to a centralized search engine, but it uses its own friendships to look for, in a decentralized manner, a node with the desired service, by contacting its friends and the friends of its friends. In this scenario, we aim to evaluate the impact of several strategies for link selection in order to select an optimal set of friendships to limit the use of computational resources needed for the search operations.

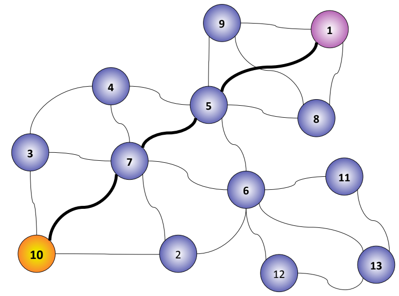

Fig. 1: Decentralized search

B. Key aspects of Network Navigability

In the past years, the problem of network navigability has been widely studied. As defined by Kleinberg [16], a network is navigable if it “contains short paths among all (or most) pairs of nodes”. Several independent works, such as [17] and [18], formally describe the condition for navigability: all, or the most of, the nodes must be connected, i.e. a giant component must exist in the network, and the effective diameter must be low. In other words, the greatest distance between any pairs of nodes should not exceed $log_2(N)$, where N is the number of nodes in the network.

When each node has full knowledge of the global network connectivity, finding short communication paths is merely a matter of distributed computation. However, this solution is not practical since there should be a centralized entity, which would have to handle the requests from all the objects, or the nodes themselves have to communicate and exchange information among each other; either way a huge amount of traffic would be generated.

Nevertheless, starting from the Milgram experiment [6], Kleinberg concluded that there are structural clues that can help people to find a short path efficiently even without a global knowledge of a network [7] [19]. This means that there are properties in social networks that make decentralized search possible. Let us suppose to have a network as represented in Figure 1, where node 1 wants to get access to the information owned by node 10 (1 doesn’t know where the information is located); obviously the optimum path leads through nodes 5 and 7. However, node 1 has three possible paths to choose from and only knows few information about its neighbors: the property that will guide node 1 to select node 5 as a next hop is that node 5 has a high degree of centrality, i.e. it has many connections. As such, node 5 represents then a network hub, i.e. a node that is connected to many other nodes. The ability for a node to quickly reach a network hub is assured by the existence of network clusters where nodes are highly interlinked: this characteristic is assured with high value of the local clustering coefficient, described by Watts and Strogatz [20], and is calculated for each node in a network. It measures how close the neighbors of a node are to being a clique, i.e. a complete graph, and it is calculated as follows:

$$C_{local}(n) = \frac{2 * e_n}{k_n * (k_n - 1)} \tag{1}$$

where k_n represents the number of neighbors of the node n and e_n is the number of edges among the neighbors.

Still, node 5 needs some additional hints in order to choose node 7 over node 6, since both of them have the same degree. This characteristic is the node similarity, an external property to the network, derived from some additional information about the nodes. In the SIoT, node similarity will depend on the particular service requested and on the types of relationships involved.

The problem of global network navigability is then shifted to the problem of local network navigability, where neighboring nodes engage in negotiation to create, keep or discard their relations in order to create network hubs and clusters.

C. Selection of network links

As described in Section II-A, objects can create, through the mimic of their owner’s behavior, several types of relationships. Other types of friendships could be added in the future, leaving to the node the hard work to cope with a huge number of connections. To make the service search process more efficient and scalable, we propose five heuristics to help the nodes in the process of selection of the best set of friends.

At first, a node accepts all the friendship requests until it reaches the maximum number of connections allowed - N_{max}. This parameter is intended to limit the computational capabilities a node needs to resolve a service search request. Then, to manage any further request, a node sorts all the friendships and the new request based on one of the following strategies:

1) A node refuses any new request of friendships so that the connections are static.
2) A node sorts its friends in decreasing order of their degree, with the aim to maximize the number of nodes it can reach through its friends, i.e. to maximize the average degree of its friends.
3) A node sorts its friends in increasing order of their degree, with the aim to minimize the number of nodes it can reach through its friends, i.e. to minimize the average degree of its friends.
4) A node sorts its friends in decreasing order of their common friends, with the aim to maximize its own local cluster coefficient.
5) A node sorts its friends in increasing order of their common friends, with the aim to minimize its own local cluster coefficient.

From the so-constructed rank list, a node accepts as its friends only the first N_{max} nodes. If the friendship with a

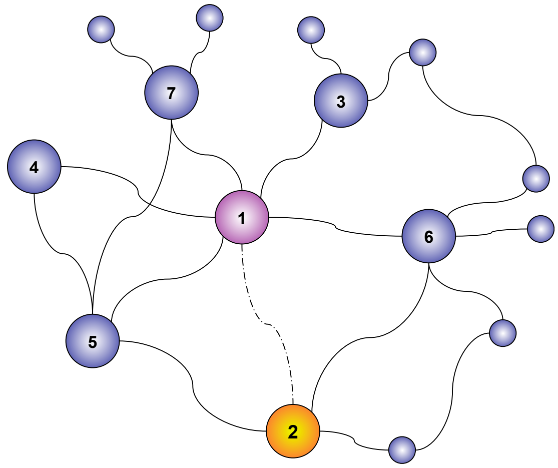

Fig. 2: Selection of network links

node is already active then nothing happens, otherwise a new friendship is created and the node with the lowest value is discarded.

Let us consider a network example, as shown in Figure 2, where the maximum number of connections is set to $N_{max} = 5$ and let us suppose that node 2 sends a friendship request to node 1 (dashed line). Since node 1 has already reached N_{max} connections, the decision on this request will depend on the implemented strategy. If node 1 implements strategy 1, it will simply refuse the request; with strategy 2, node 1 checks the degree of all its friends and of node 2 and then it terminates the relationship with node 4, which has only one more friend, in order to accept the request from node 2 (3 friends). In the same way, using strategy 3, node 1 terminates the relation with node 6, which has N_{max} connections, and accepts the request. With strategy 4, node 1 compares the common friends among its friends and with the requester node and discards the node 3 with which it has no common friends. In a similar way, with strategy 5, node 1 discards the relation with node 5 to which it has the highest number of common friends.

IV. Experimental evaluation

A. Simulation setup

With this simulation analysis, we want to study the impact of each of the proposed strategies on the objects' network navigability.

To analyze the navigability of a SIoT network, we would need information about the requests of establishing new relationships the objects would receive on the basis of their profile, settings and movements. And we would need this information for huge numbers of real objects. Even if some platforms already exist that implement the SIoT paradigm, such as [21], this data is not available to date as real applications have not been deployed yet. For this reason we had to adopt an alternative solution to test our heuristics as follows:

1) first we analyze a social network of humans;
2) from this, we extract the information needed to build the social network of objects;
3) in the next stage, we extract the characteristics of this network and use these to run a model that generates synthetic networks with similar properties;
4) and finally we apply the strategies described previously and analyze the results.

TABLE I: Parameters of Brightkite, SIoT network and Barabási-Albert model

	Brightkite	SIoT network	BA model
Nodes	12275	14557	15000
Number of Edges	39515	67363	75000
Average Path Length	4.570	4.534	3.905
Average Clustering Coefficient	0.247	0.375	0.255
Diameter	14	14	6
Average Degree	6.631	9.808	10
Giant Component	84.32%	93.45%	100%

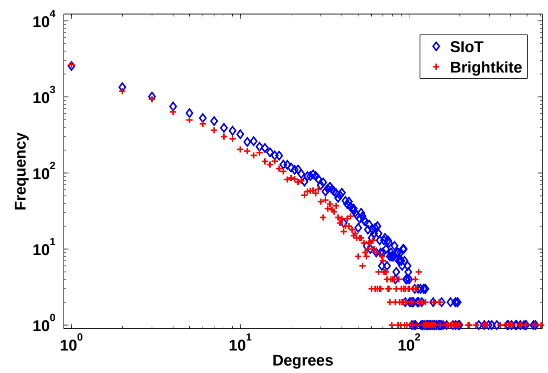

Fig. 3: Degree distribution for Brightkite and SIoT

For the first step we relied on the real dataset of the location-based online social network Brightkite obtained from the Stanford Large Network Dataset Collection [22]. This dataset consists of more than 58k nodes and more than 200k edges, so in order to better analyze its properties and compare them to synthetic data, we consider only the nodes enclosed between Atlanta and Boston for a total of approximately 12k nodes and 40k nodes. However, the output of the Brightkite dataset is a trace of the position of humans and of their relationships; since we are interested in the relationships of the objects we have extended it as follows (step 2): starting from the scaled network, we suppose that every person carries at least one smart object, for example a smartphone, so when they get in touch with their friends their objects also come into contact and have then the possibility to create a SOR. In a similar way, we also simulate the creation of CWOR and CLOR. The resulting SIoT network has around 14.5k nodes and 67k edges. The parameters of the two networks, obtained from Gephi [23], are showed in Table I, while the node distribution is shown in Figure 3 for Brightkite (in red) and SIoT (in blue) networks.

Both networks comply with the condition for network navigability: at global level, there is a giant component and the average path length is low; at local level, we can observe how the nodes are highly interlinked, thanks to the high values of the clustering coefficients, and the networks have a scale-free

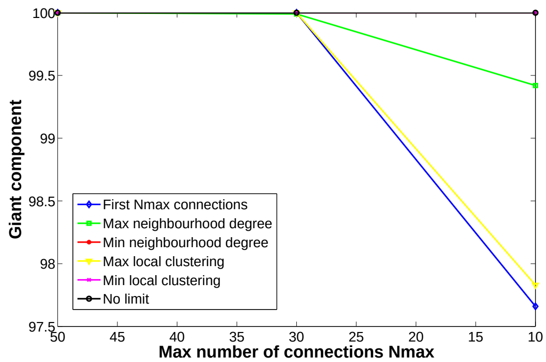

Fig. 4: Giant component for all the strategies

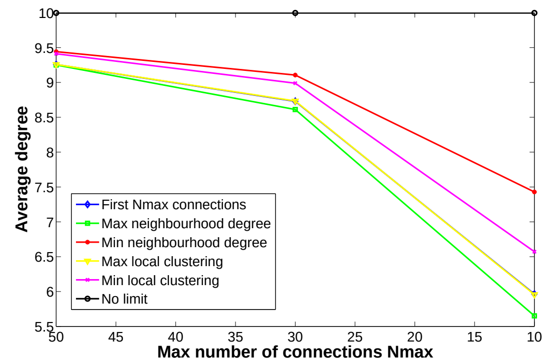

Fig. 5: Average degree for all the strategies

degree distribution thus indicating the existence of hubs.

Moreover, it is important to point out that even if the SIoT is expected to have a shorter average path length with respect to classical social networks, this does not happen since the new relationships are due to CLORs and CWORs that are indeed short range; however, for the same reason, it is possible to observe a 50% increment of the average local clustering.

To generate and analyze similar networks, we rely on the Barabási-Albert model [3], which is able to generate scale-free networks based on preferential attachment. Starting with a small number of nodes, at each step, it adds a new node with m edges (m is a parameter for the model) linked to nodes which are already part of the system. The probability p_i that a new node will be connected to an existing node i depends on its degree k_i, so that $p_i = k_i/(\sum_j k_j\ )$ leading to the name preferential attachment. The results of this model, using 15k nodes, connecting each node to $m = 5$ other nodes and averaged over 5 runs, are shown in Table I, and it can be observed that it represents a good approximation for the real scenario.

B. Simulation results

This section describes the simulation results in terms of giant component, average degree of connections, local clustering and average path length for all the heuristics described previously. Due to their complexity, we decided to run the simulations considering a maximum number of connections for a node equals to $N_{max} = 50, 30, 10$ friends. The results show this approximation is adequate to understand the behavior of the network.

Figure 4 shows the percentage of the giant component for all strategies. It is important to note that if we try to minimize the neighborhood degree or the local clustering, we can always achieve a giant component which includes all nodes. This happens due to the fact that, when a node with N_{max} connections receives a friendship request from a low connected node, it will always accept it to the detriment of a node with higher connectivity which has high probability to remain connected to the network. Moreover, we can observe that when using the strategy 1, 2 or 4, the dimension of the giant component naturally decreases with the reduction of the N_{max} value, thus making the network not fully navigable. In the case of using the strategy 2, a node connected to other nodes with N_{max} friends will not accept any other relation request, similarly to a node in a near-clique in strategy 4. Furthermore, we also want to point out, that with strategy 2 and 4 a node can not refuse or discard relationships if this action is going to isolate a node; in this way, we can achieve larger giant component and we do not have isolated nodes but at least isolated couples of nodes.

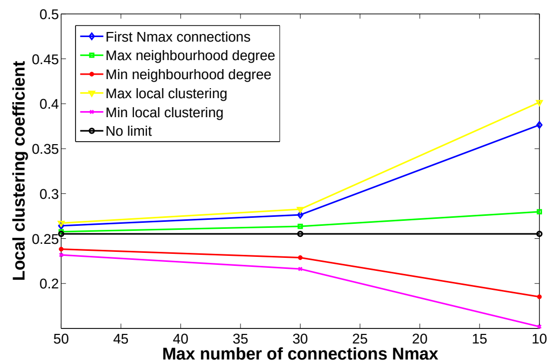

Fig. 6: Local cluster coefficient for all the strategies

From Figure 5 we can observe how the average degree changes with different strategies. Strategy 3 tries to equalize the number of friendships between the nodes, resulting in a higher number of relationships in the network and consequently a higher average degree. Similarly, strategy 5 discards the nodes with higher local cluster coefficient, to connect with nodes with low values. Yet, since the local cluster coefficient is not directly connected to the number of friends, the average degree is lower than in strategy 3. Strategy 2 achieves the lowest average degree due to the fact that the resulting network has a core of high degree nodes, with N_{max} friendships, and highly interconnected between themselves. These nodes hardly accept any new friendship, leaving many nodes with a low degree.

Figure 6 shows the local cluster coefficient. Strategy 4 and 5 exhibit the highest and lowest value respectively, since they are designed to achieve these results. Strategy 1 has a high value due to the triad formation step in the model and to the fact that there is not further rearrangement of relationships after these has been created; this effect is even stronger when the number of maximum connections is decreasing. Strategy 2 achieves a higher value than the model since the core nodes in the network are highly interconnected. It is important to point out the behavior of the local clustering coefficient for Strategy 3: it has a lower value than the model and decreases with N_{max}. This is a result of the equalization of the number of friendships, leading to a high average degree and easily destroying the triad formation step in the model.

V. Global vs local navigability

In this section, we show how the different strategies impact on the navigability of the network both at global and local levels.

Figure 7 shows the average path length when the nodes have global knowledge of the network. This means that every node is able to find the best path to reach its destination. In particular, we can observe that for N_{max} set to 50 friends all the strategies perform around the same; however, if we reduce the number of friends allowed, some differences emerge: strategies 3 and 5 provide shorter paths than the others. This is due to the fact that these strategies manage to create many long distance relationships. On the other hand, strategy 4 has the worst performance for the exact opposite reason: nodes are too close to create a clique, i.e., a subset of nodes with a full mesh topology, and have difficulties reaching other nodes; similar reasons also hold for strategies 1 and 2.

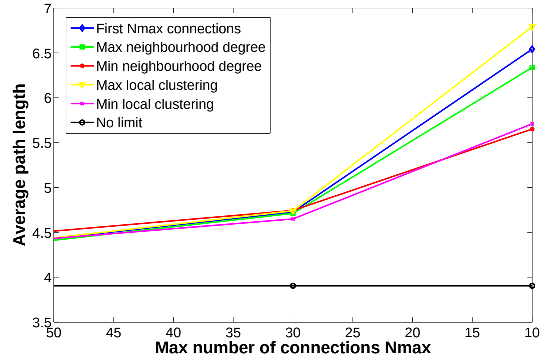

Fig. 7: Average path length for all the strategies with global knowledge of the network

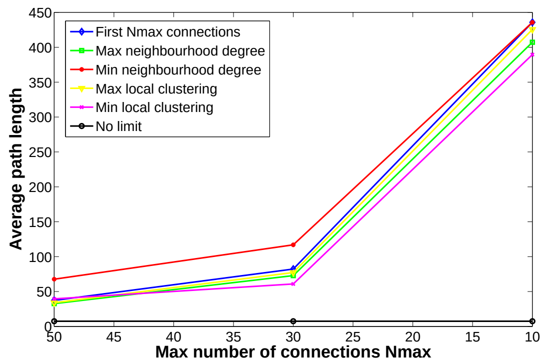

Fig. 8: Average path length for all the strategies with local knowledge of the network

However, as we said in Section III-B, we are interested in local navigability, i.e., in the ability of each node to reach the destination making use of only local information. To this we consider the following straightforward local routing approach. The scenario is of object A that wishes to communicate with node B. The first task is to check whether it has a direct connection with it, i.e., B is among its friends. If not, A asks to its friends with the highest connectivity degree, say X, to find a route to B. X then repeats the process till B is reached. Figure 8 shows the results, in terms of average path length, when the nodes only exploit local knowledge of the network. This figure shows that in the network generated with the Barabási-Albert model, without applying any of the proposed strategies, the average number of hops is around 7.5, while the best strategy needs an average of around 32 hops to reach the destination when we consider N_{max} equals to 50. This value is unthinkable if applied to a real network; however, these simulations do not take into account three fundamental aspects:

- we have considered all the possible pairs of nodes to be uniformly distributed over the network; however, it has been proved that friends share similar interests (bringing to the homophily phenomenon [24]), 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;
- we have not considered node similarity for the discovery operations: indeed, in our simulations, nodes try to reach their destination using only information about the degree of their neighbors. However, external properties could be used to select the right nodes (among the available friends) to which ask for the desired service. One of these properties is the profile of the friend involved, its trustworthiness [25], and the type of relationship that links it to the requester;
- we have not discussed about delivery of the service: depending on how the SIoT model is implemented, the service can be delivered either directly relying on the communication network (non-overlay structure) or through the friends that discovered the service, i.e. the social networks is used to transmit the service on top of the existing transport network (overlay network). In the latter case, the average path length is still an important indicator, however a longer path does not necessarily mean a higher end-to-end delay, since the delay is influenced by several factors such as the congestion of the nodes.

The analysis of the average path length is quite surprising also for another reason. As we expected, when N_{max} is equal to 50 friends, the best strategies are the 2nd and the 4th, which try to maximize the parameters for local navigability, namely the degree and the local clustering coefficient, respectively. But, if we decrease N_{max}, the best strategy becomes the number 5, in complete contrast with Kleinberg's findings, as it tries to minimize the local clustering. To understand this behavior, we analyze the degree distribution of the network, as shown in Figure 9 for strategy 2; we analyze only this strategy for simplicity but the same considerations hold for the others as well. The red + mark shows the behavior of the network without any limit for the number of friendships, while the green square, the blue diamond and the purple star refer

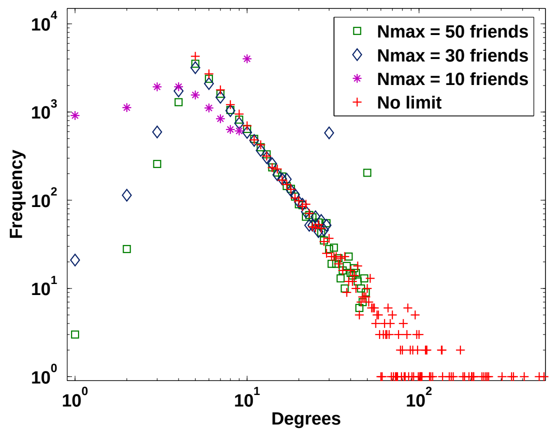

Fig. 9: Degree distribution for different value of N_{max}

to the case of N_{max} set to 50, 30 and 10 friends respectively.

Without any strategies to select the friendship, the degree distribution resembles a perfect power law. If we set a maximum number of friendships we can notice two different effects that make the distribution deviates from a power law: firstly, there are a certain number of nodes with few connections, less than four, that have more difficulties to reach the rest of the network: the lower N_{max} the greater the number of isolated nodes. Secondly, there are a lot of nodes that reach the maximum number of friends allowed: once again, the lower N_{max} the greater the number of nodes with N_{max} connections. With N_{max} equals to 10, the degree distribution is not a power law anymore, while in the other scenarios, it still follows a power law for the interval $[5, (N_{max} - 1)]$.

In particular, this last aspect deeply impacts on the navigability of the network. When a node has to choose which other node deliver a message to, its choice is driven by the degree of its neighbors: if the networks has too many hubs with the same degree, the nodes have no clues to select the next hop. For this reason, when we reduce N_{max}, the properties for local navigability no longer apply and strategies that perform better at global level, start to perform efficiently even at local level. As a result of this analysis, we have introduced a variant in the strategies so that N_{max} varies during the network life based on two different aspects: the total number of nodes in the network and the number of nodes that actually reach N_{max} friends; in other words, we want to monitor the percentage of hubs in the networks. To this, we need to constantly know both the number of nodes in the network and how many nodes have already reached N_{max} connections. These values are related to global statistics, but can be easily computed by the server; in the SIoT scenario, objects continuously communicate with the server in order to update their profile, send the data, look for information and so on, and then have the possibility to retrieve these statistics.

In particular, N_{max} increases of 10% when:

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

so that x represents the maximum percentage of hubs in the network, while y represents the threshold for a node to become a hub. It is then possible to regularly check the behavior of the network through the setting of these two parameters and modify them to directly adjust the navigability of the network.

TABLE II: Hubs percentage for all the strategies with different threshold

	No Limit	First N_{max} friends	Max degree	Min degree	Max CC	Min CC
y	x					
100%	0.23%	1.36%	1.49%	1.05%	1.37%	1.17%
90%	0.3%	1.71%	1.8%	1.33%	1.67 %	1.54%
80%	0.41%	2.23%	2.28%	1.67%	2.15%	1.97%
70%	0.55%	2.81%	2.88%	2.23%	2.69%	2.53%
60%	0.72%	3.62%	3.78%	3.01%	3.47%	3.29%

We then studied the behavior of the networks obtained with the Barabási-Albert model, with or without applying the proposed strategies, to understand when a node can be considered a hub and how many hubs are necessary in a network to maximize network navigability in terms of the average path length. Table II presents the best combinations x-y for the different strategies, considering only the scenario where N_{max} is set to 50 friends, that is the only scenario where Kleinberg's findings still hold; for the network without any strategies, we take into account a maximum number of friends equals to 110, which represents the intersection of the power law with the degree axis.

If we consider the "no limit" scenario, it is clear that the number of hubs required for the network to be navigable is really low, less than 1%. Moreover, if we relax the condition for a node to become a hub, namely we lower y, the concentration of hubs in the network rapidly increases. As expected, strategies 3 and 5, have the lowest number of hubs since their goal is to distribuite connections among all the nodes, while the aim of strategy 2 is to maximize the connections of a nodes and then it has naturally the highest concentration of hubs.

We then decided to analyze the behavior of the network considering only strategies 2 and 5, because, as proved in Figure 8, they show the best performance in terms of average path length when using only local information when N_{max} is set to 50 or lower than 30 nodes, respectively.

Figure 10 shows the average path length for different values of the maximum percentage of hubs x in the network. By reducing the number of hubs, the performance of the network increases as suggested by Kleinberg. However, the choice of the best strategy is still an important issue: strategy 2 outperforms strategy 5 even with a higher threshold y. In general, the lower the percentage of hubs or the threshold for a node to be a hub, the lower the average path length. If we relax too much these parameters, we reach the scenario "no limit". On the other hand, if they are set with too stringent values, we fall again in the scenarios with a fixed N_{max}.

Finally we present the maximum number of friends reached by a node for several combinations x and y in Figure 11. We can observe that, even if strategy 2 has the lowest average path length, the hubs created using this strategy reached a higher number of friends with respect to the one obtained using strategy 5. However, in this case, the threshold value has a greater impact on the creation of hubs highly

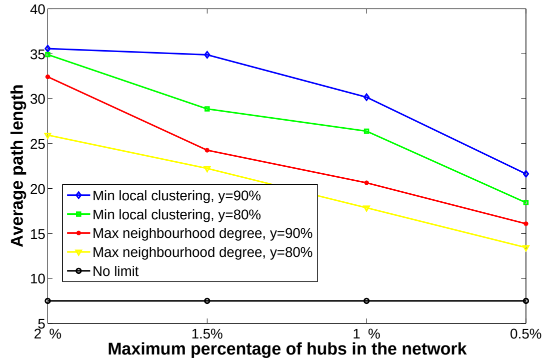

Fig. 10: Average path length for different values of the maximum percentage of hubs in the network

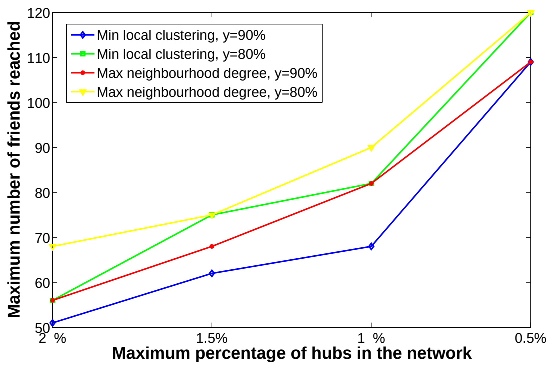

Fig. 11: Maximum number of friends reached for different values of the maximum percentage of hubs in the network

connected. To avoid this problem, we could allow only nodes with high computation capabilities, such as vehicles or smart devices, to become hubs.

VI. Conclusion

This paper addresses the issue of link selection in the SIoT, where objects establish friendship links each other creating a social network of objects. We firstly analyze network navigability in SIoT networks through simulations, as it is important for service discovery; secondly, we propose some heuristics for local link selection that have different impact in the network structure in terms of giant component, average degree and local clustering. As a result, when the network has too many hubs, selecting the friends that minimize the local neighbor degree is the approach that allows for reaching the best global network navigability. However, all these approaches have a bad local navigability, suggesting the adoption of more powerful friendship selection strategies. We then propose an approach to dynamically adjust the threshold in the number of connections on the basis of the number of hubs in the network.

References

[1] D. Zhang, L. Yang, and 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.

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

[3] B. Ostermaier, K. Romer, F. Mattern, M. Fahrmair, and W. Kellerer, "A real-time search engine for the web of things," in *Internet of Things (IOT), 2010*, 2010, pp. 1–8.

[4] K.-K. Yap, V. Srinivasan, and M. Motani, "Max: human-centric search of the physical world." in *SenSys*. ACM, 2005, pp. 166–179.

[5] L. Atzori, A. Iera, G. Morabito, and M. Nitti, "The social internet of things (siot)–when social networks meet the internet of things: Concept, architecture and network characterization," *Computer Networks*, vol. 56, no. 16, pp. 3594–3608, 2012.

[6] J. Travers, S. Milgram, J. Travers, and S. Milgram, "An experimental study of the small world problem," *Sociometry*, vol. 32, pp. 425–443, 1969.

[7] J. Kleinberg, "Navigation in a small world," *Nature*, vol. 406, p. 845, 2000.

[8] M. Nitti, L. Atzori, and I. P. Cvijikj, "Network navigability in the social internet of things," in *Internet of Things (WF-IoT), 2014 IEEE World Forum on*, March 2014, pp. 405–410.

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

[10] E. A. K. amd N. D. Tselikas and A. C. Boucouvalas, "Integrating rfids and smart objects into a unified internet of things architecture," *Advances in Internet of Things*, vol. 1, no. 1, pp. 5–12, 2011.

[11] L. Atzori, A. Iera, and G. Morabito, "Siot: Giving a social structure to the internet of things," *Communications Letters, IEEE*, vol. 15, 2011.

[12] M. Nitti, R. Girau, L. Atzori, A. Iera, and 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*, 2012, pp. 18–23.

[13] H. Wang, C. C. Tan, and Q. Li, "Snoogle: A search engine for pervasive environments," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 21, no. 8, pp. 1188–1202, 2010.

[14] A. P. Fiske, "The four elementary forms of sociality: framework for a unified theory of social relations," *Psychological review*, vol. 99, pp. 689–723.

[15] L. E. Holmquist, F. Mattern, B. Schiele, P. Alahuhta, M. Beigl, and H.-W. Gellersen, "Smart-its friends: A technique for users to easily establish connections between smart artefacts," in *Proc. of the 3rd international conference on Ubiquitous Computing*, ser. UbiComp '01. Springer-Verlag, 2001, pp. 116–122.

[16] J. Kleinberg, "Small-world phenomena and the dynamics of information," in *In Advances in Neural Information Processing Systems (NIPS) 14*. MIT Press, 2001, p. 2001.

[17] M. Bogu, D. Krioukov, and k. claffy, "Navigability of Complex Networks," *Nature Physics*, vol. 5, no. 1, pp. 74–80, Jan 2009.

[18] L. A. N. Amaral and J. M. Ottino, "Complex networks. augmenting the framework for the study of complex systems," *European Physical Journal B*, vol. 38, pp. 147–162, March 2004.

[19] J. Kleinberg, "The small-world phenomenon: an algorithmic perspective," in *Proc. of the thirty-second annual ACM symposium on Theory of computing*, ser. STOC '00. New York, NY, USA: ACM, 2000, pp. 163–170. [Online]. Available: http://doi.acm.org/10.1145/335305.335325

[20] D. J. Watts and S. H. Strogatz, "Collective dynamics of small-worldnetworks," *nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[21] R. Girau, M. Nitti, and 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.

[22] J. Leskovec, "Stanford large network dataset collection." [Online]. Available: http://snap.stanford.edu/data/

[23] M. Bastian, S. Heymann, and M. Jacomy, "Gephi: An open source software for exploring and manipulating networks," 2009. [Online]. Available: http://www.aaai.org/ocs/index.php/ICWSM/09/paper/view/154

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

[25] M. Nitti, R. Girau, and L. Atzori, "Trustworthiness management in the social internet of things," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 26, no. 5, pp. 1253–1266, May 2014.