Searching the Social Internet of Things by Exploiting Object Similarity

Michele Nitti, Virginia Pilloni and Daniele D. Giusto
Department of Electrical and Electronic Engineering - University of Cagliari, 09123, Cagliari, Italy
e-mail: *michele.nitti, virginia.pilloni@diee.unica.it, ddgiusto@unica.it*

***Abstract*—One of the major issues of Internet of Things (IoT) is discovering which objects are able to execute specific services and provide the required data for applications to be correctly accomplished. In recent years, the Social IoT (SIoT) paradigm, where objects establish social-like relationships, has become popular as it presents several interesting features to improve network navigability and implement efficient discovery methods. This is due to the intrinsic capability of a SIoT network to come to a solution for a query and quickly reach the required object with a small number of hops, using object friendships. In this paper, a SIoT-based object discovery mechanism is proposed. The novelty of this algorithm is that the next hop to query is chosen based on two properties: one that is intrinsic to the network and is based on object friendships, and an external one that takes into account the similarity between the object and the query. Simulation results prove that a correct balance between the two introduced properties is needed and that the proposed algorithm is able to efficiently solve queries, independently of their popularity and complexity.**

***Index Terms*—Social Internet of Things, object discovery, social networks**

I. Introduction

The pervasive spread of objects connected to the Internet is driving an exponential growth of Internet of Things (IoT) systems. The number of connected objects has already exceeded the world population, and it is expected to increase to 50 billion devices by 2020 [1]. The massive amount of data flowing though IoT networks poses a big issue related to their management, particularly with reference to the discovery of objects that are able to provide data by executing specific services [2].

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. Additionally, nowadays the interaction model is based on humans looking for information provided by objects (human-object interaction), but in the IoT near-future this model will quickly shift to the object-object interaction, where objects will look for others to provide complex services for the benefit of the humans, increasing the number of queries. 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.

One of the major steps in this direction has been made with the introduction of the Social Internet of Things (SIoT) paradigm, which is based on the idea that every node is an object capable of establishing social relationships with other things in an autonomous way according to rules set by the owner [5][6]. A SIoT network is expected to boost the discovery, selection and composition of services and information provided by IoT objects. The SIoT vision is based on the assumption that objects are more likely to find the required service if they look for it among their friends. This is confirmed by the observation of human social networks: according to a recent Facebook research, the average *degree of separation* among their 1.59 billion active users is 3.57, i.e. each Facebook user is connected to every other user by an average of three and a half other users [7].

As proven in [8][9], some SIoT properties ensure that it is possible to find a short path efficiently even without a global knowledge of the network, i.e. in a distributed way. Several works have analyzed the possible strategies to drive the objects to select the appropriate link for the benefit of overall network navigability [10] [11], however, to the best of our knowledge, the literature is still missing a routing algorithm to help the objects to discover the friend(s) which can provided the desired service(s).

In this paper we propose a decentralized algorithm for the discovery of objects that can provide specific application services, based on a SIoT network. In particular, for the choice of the next object to query, we focus on two properties: i) the *degree centrality*, which is intrinsic to the network and expresses how much an object is connected to the rest of the network; ii) the *object similarity* to the queried application, which is an external property with respect to the network characteristics, and defines how much the object is similar to the query requirements.

The rest of the paper is organized as follows. Section II provides a review of related works, while Section III introduces the problem. The proposed approach is described in Section IV. Finally, Sections V and VI present simulation results and conclusions.

978-1-5090-4130-5/16/$31.00 ©2016 IEEE

II. Background

A. Social IoT

The idea of using social networking elements in the IoT to allow objects to autonomously establish social relationships has gained 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 [5], [6].

Without losing the generality, in this paper we refer to the Social IoT model proposed in [12] (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. Node and Service Discovery in IoT

Searching for objects, data and services in the IoT is a crucial challenge especially in real-time environments [13] [14]. Several approaches for real-time search have been proposed in the literature, but none of them is offering a complete and satisfactory solution yet. For instance Snoogle/Microsearch [15] [16] and MAX [4] only perform local searches without taking into account the global domain; Global Sensor Networks (GSN) [17] supports searches on static metadata, whereas Dyser [3] 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. Object discovery through social tools has been proposed in [18] and in [19] where object 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 [10] or local [11] network navigability.

III. Problem Definition

Efficient discovery is made possible in the SIoT thanks to its network navigability, which is based on the properties studied by the sociologist Stanley Milgram about the small-world phenomenon [22]. As defined by Kleinberg, a network is navigable if it "contains short paths among all (or most) pairs of nodes". In other words, the greatest distance between any pairs of objects should not exceed $log_2(N)$, where N is the number of objects in the network.

In SIoT systems, objects inherit some capabilities of the humans and mimic their behavior when looking for new friends, i.e. they become friends with each other based on their common interests, as it happens for example with the social and co-work relationships. Thanks to this, SIoT networks respect the condition of navigability in the same way human networks do.

Despite this, the search of data from sensors and real-world entities is still an open issue. Indeed, when each object has full knowledge of the global network connectivity, finding short communication paths is merely a matter of distributed computation [23]. 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 objects themselves have to communicate and exchange information among each other [24]; either way a huge amount of traffic would be generated.

Nevertheless, starting from the Milgram experiment [20], 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 [8] [9]. 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 object 1 wants to get access to the information owned by object 9 (1 doesn't know where the information is located). The first property that will guide object 1 to select object 4 as next hop is its high degree centrality, i.e. the next hop has many connections and, as such, it represents a network hub, namely an object that is connected to many other objects. Still, object 4 needs some additional hints in order to choose among objects with the same degree, as it is the case when it has to choose between object 5 and 6: these hints can be found in some external property to the network, derived from some additional information about the objects. For this reason, in this paper a second property is proposed, which is called object similarity. Object similarity can encompass all the properties, characteristics and functionalities through which an object can

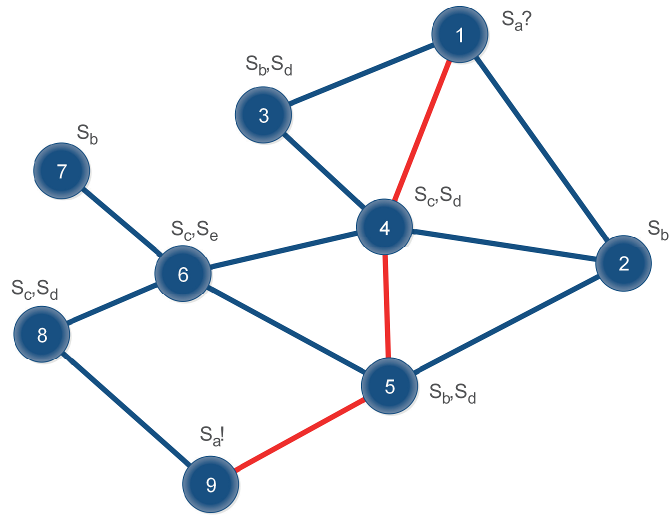

Fig. 1. Distributed object discovery

be described, and defines how similar an object is to the query requirements. Consequently, object 4 will choose object 5 over object 6 because its services, namely S_b and S_d, are more similar to the query, S_a, than the services S_c and S_e of object 6.

In the next Section, the proposed model and discovery process are described in details.

IV. Object Discovery Model

The reference scenario considered in this paper is that of a SIoT-based system. The network is modeled as an Undirected Graph (UG) $\mathcal{G} = \{\mathcal{O}, \mathcal{E}\}$, where the vertices represent the N objects $\mathcal{O} = \{o_i\}$, while the links are described by the set of edges $\mathcal{E} = (e_{ij})$, where each edge represents a social connection between a couple of objects. Finally, let $\mathcal{F}_i = \{o_j \in \mathcal{O} : o_i, o_j \in \mathcal{E}\}$ be the neighbourhoods of object o_i, namely the objects that share a relation with o_i.

Whenever a new query is received by an object o_i, the application $\mathcal{A}$ requested by that query is subdivided into services $\mathcal{A} = \{s_v\}$. We now define two subsets of application $\mathcal{A}$: the subset $A^{found} \subseteq \mathcal{A}$ of the services that have already been found; the subset $A^{res} \subseteq \mathcal{A}$ of the residual services that have not been found yet.

When the discovery process starts, $A^{found} = \emptyset$ and $A^{res} \equiv \mathcal{A}$. The first step for o_i is to look for any of the services of A^{res} among the ones it can perform. If it finds any, these services are assigned to object o_i. Thus, they are added to A^{found} and deleted from A^{res}. Then, object o_i looks for the services in A^{res} among all its friends in $\mathcal{F}_i$. Any service s_v that can be provided by a friend is assigned to it, and s_v is added to A^{found} and deleted from A^{res}. If, after these first two steps, $A^{res} \neq \emptyset$, o_i has to understand which is, among its friends $o_j \in \mathcal{F}_i$, the best candidate to look for the residual services.

As outlined in the previous Section, the discovery algorithm proposed in this paper is based on two properties: degree centrality $C(o_j)$ of object o_j and object similarity $S(o_j, A^{res})$ of object o_j with respect to the queried application A^{res}. This entails that o_i chooses the friend o_j that is associated to the highest probability value $\mathbb{P}(o_j, A^{res})$ to find the residual services, defined as

$$\mathbb{P}(o_j, A^{res}) = \alpha C(o_j) + (1 - \alpha) S(o_j, A^{res}) \qquad (1)$$

Accordingly, the query for the residual services A^{res} is sent to o_j, which starts the same sequence of steps performed by o_i. The discovery process ends when $A^{res} \equiv \emptyset$.

A. Degree Centrality

The first property we want to analyze is the notion of centrality of an object o_i, which provides a peculiar information about the social network. If a node has many relationships then it represents a network hub and it is likely used by other objects to reach other nodes in the network since it has the highest probability to know an object which can provide the desired services.

The contribution of the centrality can be computed as follow

$$C_j = \frac{|\mathcal{F}_j|}{max_{o_i \in \mathcal{O}} |\mathcal{F}_i|} \qquad (2)$$

where $|\mathcal{F}|$ is the cardinality of $\mathcal{F}$.

To keep this measure in the range [0, 1], we normalize it for the maximum number of friends of an object. It is important to note that every parameter needed to compute this property can be calculated using the local knowledge of the objects. The only parameter that could require global knowledge is the maximum number of friends; however recent works, such as [10], have proposed strategies to select the friendships and then being able to set a dynamic maximum value of connections in the social network.

B. Object Similarity

As defined in Section III, object similarity is an external property to the SIoT expressed as the similarity between an object and the requirements of the query that it has to respond to [25]. In other words, since the query is expressed as the set of services included in the subset of the residual services A^{res}, object similarity reflects the likelihood that an object is able to find the services in A^{res}. Therefore, the higher the likelihood that o_j's friends can perform A^{res}'s services, the higher the value of $S(o_j, A^{res})$. As a consequence, object similarity is strictly related to the similarity between the services that an object can provide and the required services, namely the service similarity. The service similarity considered in this paper is based on the semantic similarity between services. More specifically, the semantic shortest path concept is used [25].

Application services are here represented as a Directed Acyclic Graph (DAG) where each service s_v is connected to another service s_w if s_v uses the output data provided by service s_w as an input. Accordingly, the semantic distance $d(s_v, s_w)$ between s_v and s_w is dependent on the construction of the graph and can be described as the shortest path going

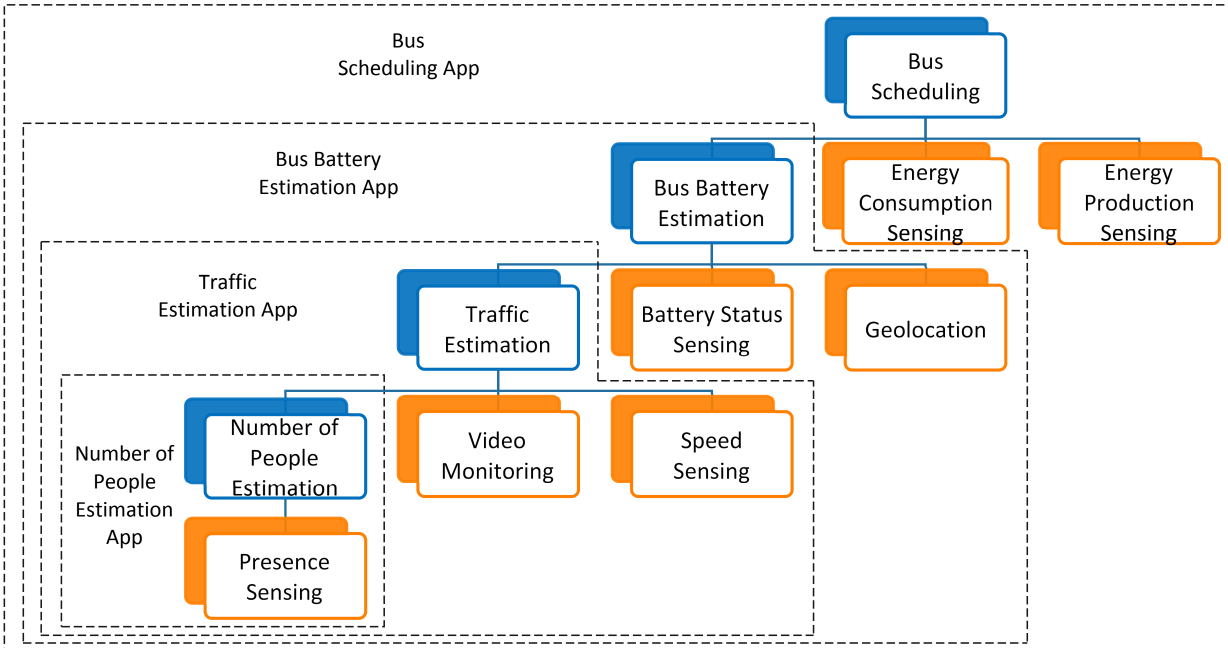

Fig. 2. Sketch of a part of the graph considered in this paper to describe applications

through an ancestor service that is common for both s_v and s_w

$$d(s_v, s_w) = ShortestPath(s_v, s_w) \tag{3}$$

Figure 2 shows an example graph, which is a part of the DAG implemented to describe the applications used to evaluate the performance of the proposed discovery algorithm (see Section V). In the example, 4 nested applications are described. Services in orange boxes represent sensing services; services in blue boxes are processing services, which need the input provided by sensing services to be executed. Referring to Figure 2 and considering, for instance, *Presence Sensing* and *Geolocation*, their closest common ancestor service is *Bus Battery Estimation*, which is 3 hops away from *Presence Sensing* and 1 hop away from *Geolocation*. Therefore, the shortest path between the two sensing services is $d(PresenceSensing, Geolocation) = 4$.

Based on the definition of distance between two services, the semantic distance $D(o_j, s_w)$ between an object o_j and a generic service s_w can be defined as the shortest path between any of the services that o_j is able to perform and s_w

$$D(o_j, s_w) = \min_v d(s_v, s_w), \qquad \forall s_v \text{ performed by } o_j \tag{4}$$

It is now possible to describe the semantic similarity between object o_j and the services in A^{res} as the complementary value of the average evaluated over all the semantic distances between all the services of object o_j and all the services included in A^{res}, normalized for the maximum possible service distance d^{max} between any couple of services represented in the application DAG

$$S(o_j, A^{res}) = 1 - \frac{1}{d^{max}} \frac{\sum_{s_w \in A^{res}} D(o_j, s_w)}{|A^{res}|} \tag{5}$$

where $|A^{res}|$ is the cardinality of A^{res}.

V. Performance Evaluation

A numerical evaluation has been conducted by using the MATLAB® tool for a wide set of scenarios, to observe the achievable performance of the model in reaching the desired service. In doing so, information about the position and mutual relationships are needed for huge numbers of real objects. This data is not available to date, as real applications involving a high number of Social Things have not been deployed yet. For this reason we resorted on the real dataset of the location-based online social network Brightkite obtained from the Stanford Large Network Dataset Collection [26] and extended it to take into account social relationships.

This dataset consists of more than 58k nodes and more than 200k edges, so to simplify the analysis, we consider only nodes in the area that is delimited by the cities of Atlanta and Boston. 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, it has been extended as follows: it is assumed that every person carries one smart object, for example a smartphone or a tablet, so when they get in touch with their friends their objects also come into contact and have then the possibility to create a SOR.

Additionally, on the basis of the mobility of the humans carrying their objects, the CWOR, CLOR, and POR relationships are created. Recall from Section II-A that CWOR and CLOR are the relationships created when objects come into contact in public and personal experiences respectively, while SOR are established when objects meet for social reasons related to their owners. The resulting SIoT network has nearly 15k

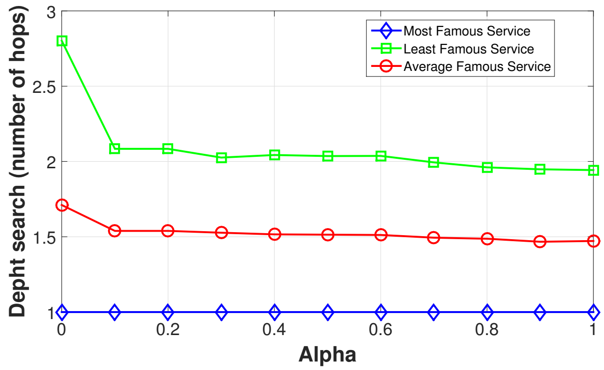

Fig. 3. Number of hops needed to perform a depht search for different values of alpha

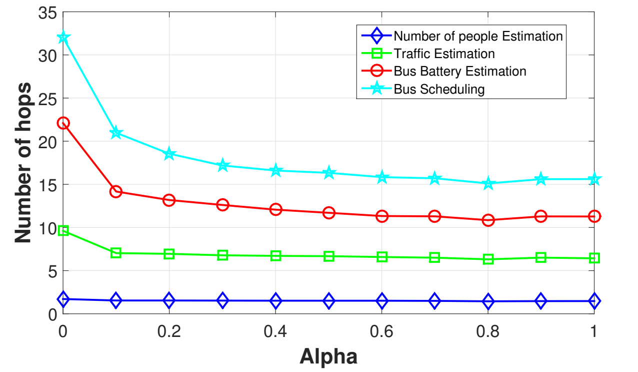

Fig. 4. Number of hops needed to find all the services to satisfy a query for different values of alpha

nodes and more than 500k edges. The number of edges in the network is higher with respect to the original dataset due to the creation of new relationships among objects.

Each node can provide three different services, whose popolarity have been assigned following a Zipf's law.

The semantic description of applications and the DAG of the services has been described using the RDF language in the Protégé tool, developed by the Stanford University. The DAG accounts for 100 applications. Each application has at least one processing service and one sensing service, for a total amount of 102 processing services and 100 sensing services. Note that the same service can be used by more than one application (e.g. *Presence Sensing*, in Figure 2 is used by all the 4 applications depicted in it). The DAG has been later deployed on a local server, and queried using the SPARQL query language in Matlab, through a client/server interface.

During the object discovery process, whenever a given object has some residual services to find, it will forward the query to the friend which has the highest probability to find *all* of them. This way, at every hop, more than one service can be find at once, instead of sending a request to different objects for each service. Figure 3 shows the number of hops needed to find a single service, when varying its popularity in the network, for different values of α. When the required service is the Most Popular Service (MFP) in the network, it can be found as fast as with a single hop, i.e., on average, every object has at least a friend which can provide that service; on the other hand, for the Least Famous Service (LFS), we need around 2 hops to find it, which is still a good result considering that the average hop length, calculated through Gephi [27] and then with a global view of the network, is 2.9. Slightly differences can be observed when increasing the α value. To better understand its behavior, we want to analyze the results for queries of different complexity while considering services with similar popularity. With reference to the example shown in the previous Section, Figure 4 shows the number of hops needed to solve the queries while Figure 5 refers to the average number of hops required to find each service in the same query. Firstly, We observe that there is a considerable difference in

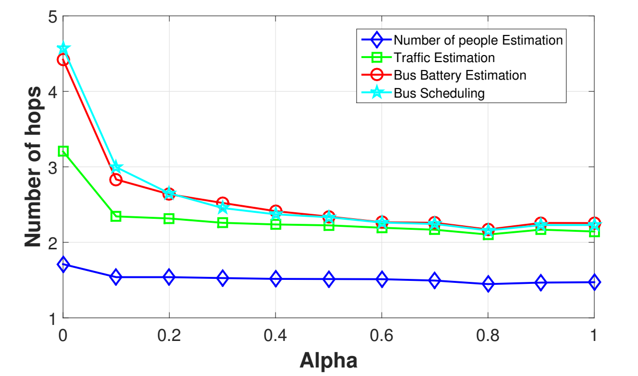

Fig. 5. Average number of hops needed to find each service to satisfy a query for different values of alpha

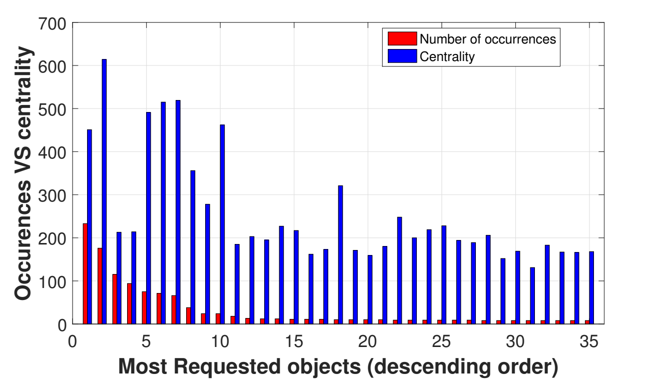

Fig. 6. Number of hops needed to find all the services to satisfy a query for different values of alpha

the number of hops when $\alpha = 0$, i.e. considering only the object similarity, and when $\alpha = 1$, i.e. making use of the degree centrality, so that, in line with Kleinberg's findings, the latter is of crucial importance to navigate the network with only local information. Secondly, independently of the query complexity, the best results are obtained with $\alpha = 0.8$; this result justifies the need of another parameter other than the degree centrality to perform more refined searches.

This conclusion is also supported by Figure 6, which shows the number of times a node is requested to find the residual services against its degree centrality. We can notice that there are objects, namely the third and fourth objects most requested, which have a very low degree centrality compared to objects with a similar number of occurrences, but that are often used by the other objects to forward the query.

VI. Conclusions

This paper addresses the issue of object discovery in the SIoT, where objects establish friendship links among each other to create a social network of objects. Whenever a new query is received by an object, it checks if one of its friends is able to perform them, otherwise it needs to choose among them, the one that has the higher probability to resolve the query. The proposed algorithm is that the next hop to query is chosen based on two properties: one that is intrinsic to the network and is based on object friendships, and an external one that takes into account the similarity between the object and the query.

As future works, we plan to deeply study if object discovery can be improved through the use of other parameters, both intrinsic and external. For example, the problem of semantic distance among services is that it is dependent on the construction of the graph, so we plan to find similar parameters which are less sensitive on the used model.

Acknowledgements

This work was supported in part by Italian Ministry of University and Research (MIUR), within the Smart Cities framework (Project: CagliariPort2020, ID: SCN_00281).

References

[1] D. Evans, "The internet of things how the next evolution of the internet is changing everything. cisco white papers," 2011.

[2] M. Nitti, V. Pilloni, G. Colistra, and L. Atzori, "The virtual object as a major element of the internet of things: a survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1228–1240, 2015.

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

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

[7] S. Edunov, C. Diuk, I. O. Filiz, and S. B. Burke, "Three and a half degrees of separation," 2016. [Online]. Available: https://research.facebook.com/blog/three-and-a-half-degrees-of-separation/

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

[9] ——, "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

[10] M. Nitti, L. Atzori, and I. Cvijikj, "Friendship selection in the social internet of things: challenges and possible strategies," *Internet of Things Journal, IEEE*, vol. PP, no. 99, pp. 1–1, 2014.

[11] L. Militano, M. Nitti, L. Atzori, and A. Iera, "Using a distributed shapley-value based approach to ensure navigability in a social network of smart objects," in *2015 IEEE International Conference on Communications (ICC)*. IEEE, 2015, pp. 692–697.

[12] 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.

[13] 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.

[14] C. Campolo, A. Iera, L. Militano, and A. Molinaro, "Scenario-adaptive and gain-aware content sharing policies for cooperative wireless environments," *Computer communications*, vol. 35, no. 10, pp. 1259–1271, 2012.

[15] 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.

[16] C. C. Tan, B. Sheng, H. Wang, and Q. Li, "Microsearch: A search engine for embedded devices used in pervasive computing," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 9, no. 4, p. 43, 2010.

[17] K. Aberer, M. Hauswirth, and 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.

[18] A. Jara, P. Lopez, D. Fernandez, J. Castillo, M. Zamora, and A. Skarmeta, "Mobile digcovery: discovering and interacting with the world through the internet of things," *Personal and Ubiquitous Computing*, vol. 18, no. 2, pp. 323–338, 2014. [Online]. Available: http://dx.doi.org/10.1007/s00779-013-0648-0

[19] A. J. Jara, P. Lopez, D. Fernandez, J. F. Castillo, M. A. Zamora, and A. F. Skarmeta, "Mobile digcovery: discovering and interacting with the world through the internet of things," *Personal and ubiquitous computing*, vol. 18, no. 2, pp. 323–338, 2014.

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

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

[23] M. Anedda, G.-M. Muntean, and M. Murroni, "Adaptive real-time multi-user access network selection algorithm for load-balancing over heterogeneous wireless networks," in *Broadband Multimedia Systems and Broadcasting (BMSB), 2016 IEEE International Symposium on*. IEEE, 2016, pp. 1–4.

[24] G. Colistra, V. Pilloni, and L. Atzori, "Task allocation in group of nodes in the iot: A consensus approach," in *2014 IEEE International Conference on Communications (ICC)*. IEEE, 2014, pp. 3848–3853.

[25] A. Bernstein, E. Kaufmann, C. Bürki, and M. Klein, "How similar is it? towards personalized similarity measures in ontologies," in *Wirtschaftsinformatik 2005*. Springer, 2005, pp. 1347–1366.

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

[27] 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