A Social-Aware Approach for Federated IoT-Mobile Cloud using Matching Theory

Sara Ranjbaran[+], Mohammad Hossein Manshaei[+], and Michele Nitti[†]
[+] Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan 84156-83111, Iran
[†]DIEE, University of Cagliari, UdR CNIT of Cagliari
Emails: s.ranjbaran@ec.iut.ac.ir, manshaei@cc.iut.ac.ir, michele.nitti@diee.unica.it

***Abstract*—In the Internet of Things (IoT) scenario, the deployment of integrated environments have pushed forward the collaboration of heterogeneous devices to match wide-ranging user requirements. However, several open challenges need to be solved such as the intrinsic unreliability of IoT devices as well as the variety in users' preferences when sharing their devices. In this paper, we give a contribution by proposing a novel hybrid paradigm to support the cooperation among IoT devices and exploit their unused resources. Our solution is based on the Social IoT concept (SIoT), where objects are connected to the Internet create a dynamic social network based on the rules set by their owner. In particular, we introduce the concept of Social Mobile-IoT Clouds (SMICs), where heterogeneous devices combine their resources to serve other co-location devices requirements. In the proposed mechanism, the notion of object sociality is considered to build the required trustworthiness among devices. To this aim, we make use of a Many to Many (M–M) assignment game based on matching theory to support the cooperation among devices. Our simulation results confirm the enhancements achievement in terms of percentage of resources being successfully assigned.**

I. Introduction

Internet of Things (IoT) encompasses a large number of uniquely addressable smart devices and support diverse applications, interacting in a non-trivial way. This vision foresees a billion of everyday physical objects, equipped with different capabilities, to be connected to the Internet and enhanced with intelligence in order to make smart our environment [1]. However, the IoT includes devices with heterogeneous resources, both considering their computing and storage capabilities and the adopted communication technologies, which then rely on Cloud solutions to implement the required collaboration to run applications. Recently, fog computing [2] has emerged to address the issues of cloud-based solutions, by offering a faster processing time and a lower latency in order to meet applications' requirements. This is achieved by moving the remote data centers toward the edge of the network and deploying a large number of distributed virtualization platforms between end-user and the cloud.

Moving forward on this road, the idea of IoT devices cooperating opportunistically is gaining even more popularity. This approach, identified as Mobile Edge Computing (MEC) [3], focuses on the possibility to create local cloud by aggregating group of devices and their relevant resources to further reduce service execution time [4]. Besides, as the range of devices seeking for online services continues increasing, several substantial works have emerged in literature around designing cooperative method to utilize the potential of IoT devices, such as [5]–[8]. However, all the existing works neglect to address two important factors: the owners possessing the devices have different goals when they decide to share their devices' resources and may need some kind of incentive in order to participate in a local cloud; moreover, the IoT environment is a unreliable system, where malicious nodes represent a constant threat for a successful cooperation, especially when considering the direct exchange of information among devices.

To solve these problems, we rely on a novel approach based on the vision of social relationships among different devices. Without losing generality, we refer to the Social Internet of Things (SIoT) paradigm proposed in [9], where devices can create and manage their own relations based on the set of rules set by their owners. This is expected to increase the probability of cooperation, since considering social behavior can act as an incentive for cooperation among devices [10]. Moreover, it has been studied that a trustworthiness management based on the SIoT concept is able to improve the performance of the network by filtering out untrusted devices [11].

With regard to these considerations, in this paper we investigate the situations where several connected components combine into temporary collections of functions and services. We refer to such collections as "Social Mobile-IoT Clouds" (SMICs), that allow users to create federations of different type of resources that can be shared with other users in order to run applications. Indeed, this approach utilizes the opportunistic resources available from co-location devices, existing in the coverage of same network *edge node* (EN). The local EN, as a first access point or fog node, is designed for control and management function, while differently from fog services, the whole task execution is delegated to the devices themselves. The main role of EN is to coordinate the matching between application requests and device resources in order to maximize the user preferences.

The paper is organized as follows. Section II provides an overview on related works. Section III illustrates the reference scenario and the problem formulation. In Section IV the problem of the edge assisted Social Mobile-IoT Clouds is formulated by means of a matching optimization algorithm. The performance evaluation is reported in Section V, and conclusions and future works hints are given in Section VI.

978-1-5386-4980-0/19/$31.00

II. State of the art

This section provides a brief overview of related works on mobile cloud and fog computing as well as a brief introduction to social IoTs.

A. Mobile Cloud and Fog Computing

With recent advances in cloud computing, a new paradigm is emerged named *mobile cloud computing* (MCC). The main goal of MCC is to help mobile users by providing a rich functionality, regardless of the resource constraint of mobile devices. Its major focus is enhancing computations and storage capabilities of mobile devices by outsourcing some services to cloud data center.

In [3], the interaction between mobile devices and a fixed data center is investigated in order to provide resources for desired services. Generally, there are two main architectures for MCC: in the first category, there is a central high power data center that provide resources for mobile users. The second category encompasses cooperation-based methods, where mobile devices also can share their resources and services. In [12], a distributed solution to create local mobile clouds is presented. However, in such infrastructure-less solutions, the overhead of signaling among devices to create and maintain the local clouds is too heavy. On the other hand, the sheer limitations of central based MCC architecture are the bandwidth and latency caused from communication with remote data center, that is not acceptable for some IoT applications that need real-time services.

To overcome these restrictions, recently, cooperative-based MCC extends the cloud resources to the edge of the network. That is referred to fog computing paradigm [2]. Fog computing is expected to be a promoter of mobile cloud computing and reproduces some new applications and services. The idea of creating federation for mobile IoT clouds implemented in the network's edge node is presented in [4]. In this paper the coalition formation game and the concept of Nash-Stable solution is used in order to mange the federations. Instead, in [8] the cooperation of devices is analyzed in order to share resources among multiple sponsors. It is claimed that participation in the proposed mechanism is always in the favor of the parties. Nevertheless, the cooperation is occurred in exchange for money between different service providers. The idea of a resource coordinator elected among the mobile nodes is also proposed in [13] to manage the matching between application requests and device resources. However, the authors do not actually tackle the heterogeneity of IoT devices in terms of power, autonomy, and capabilities.

None of these papers consider the interactive scenario to evolve cooperation among agents during the time. But the mechanism proposed in this paper takes into account the social relations among devices as a factor of trustworthiness and as an incentive to cooperate.

B. Social IoT

The SIoT is a new paradigm which represents the convergence of IoT and social networking technologies. In fact, SIoT creates social networks in which things are nodes that establish social links as humans do [9] in real life. Considering several potentials of social networks within IoT domain, this concept is fast gaining the ground. For example, simplification in the navigability of a dynamic network of billions of objects; robustness in the management of the trustworthiness of objects when providing information and services; efficiency in the dynamic discovery of services and information, are among the key features in SIoTs.

According to the SIoT models, every node is an object capable of establishing social relationships with other things in an autonomous way according to rules set by the owner. The following possible types of relationships can be defined for these networks: the *Ownership Object Relationship (OOR)* is created between objects that belong to the same owner; stationary devices located in the same place create the *Co-location Object Relationship (C-LOR)*; the *Parental Object Relationship (POR)* is created between objects of the same model, producer and production batch, while the *Co-work Object Relationship (C-WOR)* is established between objects that meet each others at the owners' workplace, as the laptop and printer in the office. Finally, the *Social Object Relationship (SOR)* is created as a consequence of frequent meetings between objects, as it can happen between smart phones of people who use the same bus every day to go to school/work, people hanging out at the same bar/restaurant/gym.

The resulting social network can be shaped to be navigable, which is a fundamental feature to perform searches for specific services in effective and efficient ways. Furthermore, it enables to increase the trustworthiness of acquired service among entities [10], which might be useful to support security. However, most of the IoT devices do not have the processing and communication capabilities required for the creation and management of social relationships [14]. Accordingly, SIoT solutions envision cyber counterparts of physical objects, which we call *social virtual objects* (SVOs), running on some server virtualized in the cloud. Such an approach has several advantages but suffer from a few major problems. In fact, objects might be far away from the data center hosting the cloud, which results in long delays and inefficiency in the use of the communication resources. Without loss of generality, in this paper we suppose that the SVOs are stored in the edge nodes to reduce the latency in offering the required services [15].

III. Reference Scenario and Architecture

In this paper, we assume that objects belong to a specific entity, which could be either private, as it happens for objects owned by a user, or public, as it is the case of group of smart devices belonging to a municipality. In our scenario, we assume that the *edge node* (EN) is the first access point to the network for the devices and that each physical device has a virtual counterpart on the EN, called *social virtual object* (SVO) [16], which hosts information regarding the user's preferences, the available resources of the device, implements

the social behavior and maintains social relations with other devices.

The goal of this paper is to exploit the potential cooperation among friends devices in a local environment tied by social relationships, the so-called Social Mobile-IoT Clouds (SMICs), in which the network edge node works as a coordinator to manage such local clouds. The main idea behind the SMICs concept is that, the heterogeneous co-located objects can cooperatively share their resources to distributively run their applications, rather than using cloud/fog services and with a lower latency and cost. Moreover, considering social interaction among devices can improve the reliability of services and also provide incentive for devices to cooperate with each other [10]. In particular, the edge node is able to identify the atomic tasks required to run integrated IoT applications, and then decides how to allocate these tasks among the friends devices in a SMICs taking into account the relationships among devices to enhance the trustworthiness of the resulting application. The edge node connects devices with available resources with those who are in need of resources to run their applications, in order to satisfy the user requirements and their corresponding tasks (e.g. computing, storage, sensing or actuation). In the case that there are not enough resources to satisfy the application requirements, the remote cloud acts as a backup to guarantee the task execution.

Figure 1 shows an example of the proposed scenario, with different devices belonging to both private and public entities: the dashed clouds represent all the devices of a single entity, the blue lines among SVOs indicate that there is a social relation among them, while the red circles are the possible SMICs as orchestrated by the EN. Whenever an entity requires a composite IoT application, her devices send a resource request to the edge node. The EN starts the coalition formation process in order to create a SMICs, so that each request is matched with the fittest available resources based on the entity's requirements and preferences.

Once the SMICs is formed, the EN is in charge of assigning the atomic task to be offloaded and to oversee the exchange of resources. Each device can offer its resources to one or more different SMICs, as it can happen for the information provided by a sensor, which can be reused by several devices. For this reason, it is important to develop a system to appropriately match the tasks in a reliable way.

A. Notation and Problem Formulation

With reference to the scenario presented in the previous section, we consider a system with a single edge node and several mobile SMICs. The main goal of the edge node is to organize one or more SMICs in order to increase the number of services offered without relying on the Cloud. We assume that the edge node collects over a certain time frame the service requests coming from the devices in the area under its coverage and then, based on the available resources offered by the other devices, assists them in matching their available resources to the received requests.

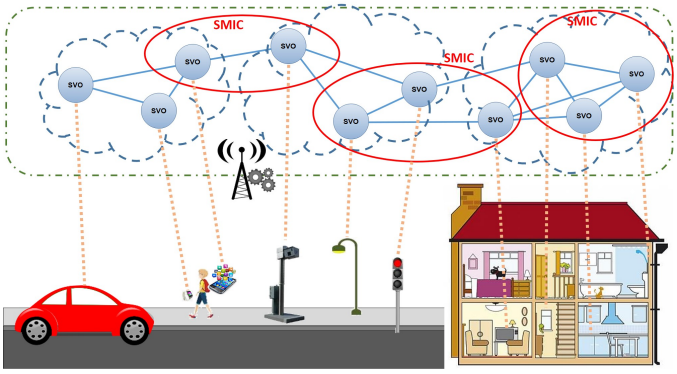

Fig. 1. Resource sharing scenario in social IoT framework.

Let us define a set of N users, $\mathcal{U} = \{U_1, U_2, ..., U_N\}$, where each user U_i (private or public) owns a set of $\mathcal{D}_i$ devices, each one of them with a virtual counterpart on the edge node, namely with its corresponding SVO. Each device d is enriched with several type of resources, $\mathcal{R}_{i,d} = R^t : t \in \mathcal{T}$, where t indicates the type of resource from the set of resource type $\mathcal{T}$. Then, the t-th type of resource belonging to d, is denoted as $R^t_{i,d}$.

Another important parameter characterizing the users is their mobility, and then the mobility of all the devices they are carrying. In this paper, we suppose that users alternate movements and waiting times, so whenever they move from one location to another, they will inform the edge node with an evaluation of their waiting time in that location. The mobility parameter is fundamental to estimate the probability that a task is performed timely; in fact, if a user moves from the coverage are of one edge node to another, also the results have to be sent to the original edge node in order to be delivered to the device requesting them, thus leading to an increase in the latency of the application [15]. For the proposed scenario, we then define for each user U_k the arrival time, a_k and the departure time d_k from a given location. Also, we assume that the users have uncertainty about the duration of the waiting time of other members. It is shown in [17] that this assumption leads to a more cooperative behavior.

Moreover, whenever the edge node has to match the received requests with the available resources, it has to take into account the energy level of each devices. To this, we consider that to each device d is associated an energy level E_d, so that the edge node can automatically discard all the devices with a energy level below a set threshold.

The creation of a SMIC is guided by the users' preferences. To this, we define a set of possible preferences as $\mathcal{P} = \{p_1, p_2, ..., p_M\}$, so that a generic user U_k can specify a vector $\mathcal{P}^k[\gamma^k_m]$ of her preferences with γ^k_m representing the weight for preference p_m assigned by user U_k and $\sum_{m=1}^{M} \gamma^k_m = 1$.

Moreover, a user can also indicates how many requests can be assigned to each of her resources at most, S. Indeed each device has a quota that refers to the maximum number of requests that it can be matched with. When sharing their own

resources, each user is considered rational, i.e., they want to maximize their own utility function.

A single user U_i can request an application a_i, which is mapped by the edge node into a set of elementary tasks requests, where $t_{i,a}^m$ represents the generic task i of type m in the application a. In the service model, for simplicity, we assume that the EN has a predefined time frame to collect all the available resources and that each request has a time frame in which the required application must be completed. Moreover, the request also specifies the maximum number of devices V that could be used to satisfy the application: this is an important parameter, since the higher the number of allowed devices, the simpler is to find a matching devices, but it is also higher the possibility that one of the them will leave the coverage area of the edge node, thus delaying the service provisioning.

IV. Matching Optimization

After collecting all information about the set of available resources, the next step is the creation of pairs of mapped devices based on the user preference $\mathcal{P}$ according to a *matching algorithm*. This will assign the number of tasks requests being served for the involved devices. In this paper, we take forward our research by proposing a model based on the matching theory for the federation formation problem. We consider many to many (M-M) matching as a kind of assignment game. The M-M assignment allows devices to connect many but different resources and resources to be assigned to many but different requests. In our setting, the matching problem is between the set of available resources and the set of applications requested by devices. The main aim of this mechanism is finding the highest performance match, in order to fulfill users objective.

We use as a starting point the approach proposed in [18] to perform M-M matching mechanism. We assume that at given point in time the EN receives n service requests and m available resources statement. As previously described, each request may consist of a set of tasks that a device d needs before a certain time τ. For example, an application required by a device may consist of several sensing and processing resources. This condition is also established for available resources, i.e., the devices may have more than one free resource at a given time. Considering this, in order to comply with the required request, a single application may require resources from more than one devices.

The goal of our algorithm is then to maximize the matrix of all possible allocations, namely $T[i,j]$, which then represents the best possible matching for the given scenario when request j is assigned to resource i. When maximizing the allocation matrix, the EN needs to keep into account a performance matrix $Q[i,j] \in [0,1]$, which expresses the performance values for the requester for each available resource based on his/her preference and it is used to weight the allocation matrix. We can then describe the optimization problem as follows:

$$
\begin{aligned}
& max \left\{ \sigma = \sum_{i=0}^{m-1} \sum_{j-0}^{n-1} Q[i,j] \times T[i,j] \right\} \\
& \text{subject to} \\
& T[i,j] \in [0,1] \quad (0 \leq i < m, \quad 0 \leq j < n) \\
& \sum_{i=0}^{m-1} T[i,j] = V[j] \quad (0 \leq j < n) \\
& \sum_{j=0}^{n-1} T[i,j] = S[i] \quad (0 \leq i < m)
\end{aligned}
\tag{1}
$$

where the vector $V[j]$ represents the maximum number of devices that must be assigned to each request j, while the vector $S[i]$ specifies the maximum number of requests that can be served by the resource i.

Whenever matching the received requests with the available resources, the EN has to take into account users' preferences, i.e. it has to compute the preference matrix $\mathcal{Q} = [Q_{i,j}]$, where each element indicate the maximum weight of preference between pairs of available resources and requests. In order to fill matrix Q, we first extract the preferences from user's profile. Then, each element $q_{i,j}$ shows how good is the matching between request j from a generic device belonging to user U_k, with her preference P^k, and the available resource i and it is calculated as follows:

$$
q_{i,j} = \mathcal{P}^k[\gamma_m^k] x F[f(i,j)] \tag{2}
$$

where $F[f(i,j)]$ is a column vector of functions of dimension M, one for each preference, that correlates request j and resource i. If a user does not have any preferences and just need to accomplish her applications, such as the case for public devices, then the weight will be calculated based only on the available time of resources.

V. Numerical analysis

In this section we analyze the performance of the proposed algorithm in terms of number of fulfilled requests. We consider a generic IoT scenario, where devices can request several application services, whose elementary tasks include sensing, actuating, computation and storage activities. We relied on the real dataset of social IoT objects accessible here[1] [19]. This dataset consists of more than 16k nodes and more than 550k links distributed over the city of Santander in Spain, so we consider a single edge node with around 1000 devices under its coverage area.

This dataset contains information regarding the objects, such as typology and the owner, their profiles (that express the set of available services and related applications), mobility and devices positions, as well as the social relations that each node can create with the others based on the rules set in the system. However this dataset only consider sensing tasks, so we need to extend it to take into account the other type of tasks; moreover, even if the devices are separated based on their

[1] http://social-iot.org/index.php?p=downloads

TABLE I
MAIN SIMULATION PARAMETERS.

Parameters	Value
Number of users	200
Average number of devices per user	5
Number of available resources	[2000-4000]
Sensing units per service request	[1-8]
Computation units per service requests	[1-15] MFLOPS
Storage units per service requests	[10 -1500] MBytes
Actuating units per service request	[1-3]
Number of sensing resource per device	[2-8]
Number of Actuation resource per device	[1-3]
Amount of RAM per device	[0.1-16] GByte
Amount of storage per device	[0.1-1024] GByte
Energy level of mobile devices	[50-4000]mAh

mobility, i.e., fixed or mobile, there is no information regarding their power sources. The extended dataset is available at the same webpage; in particular, the object profile now also specifies a category for computation and storage capabilities based on the number of cores, the amount of memory and the amount of storage as well as the set of possible actuators offered by each category of devices. Moreover, we consider that static objects are usually plugged to the grid and then have no energy consumption issues, while mobile devices have a maximum battery level based on their category and consume energy for every task solved, except for the sensing ones which can be provided by the corresponding SVO. Table I shows all the parameters for our simulations.

As previously discussed, the edge node periodically collects the available resources from the devices. Depending on how the federation process is implemented, the matching algorithm could run either periodically after the resource collection phase or as soon as a request is received, i.e. on demand. In the first case, the edge node has a clear view of all the requests that need to be satisfied and of the pool of available resources; instead, with the on demand approach, the EN provides the best matching for each received request but without taking into account if and how many requests it will receive before the next time-frame.

The analysis is conducted by using a simulation code developed in Python to study the impact of the number of requests on the percentage of served requests. In particular, we analyze the results of the proposed matching algorithm when considering the two approaches describe above, namely on demand and periodic, and with different values of available resources. To compute an accurate confidence interval, each scenario has been evaluated over 40 different runs. The results shown in Figure 2 refers to the percentage of served requests: please note that a request is considered successful only if all the tasks of the same application request have been executed inside the coverage area of an edge node.

As expected, the percentage of executed requests decrease with the number of requests since there are not enough available resources to satisfy all of them. The performance

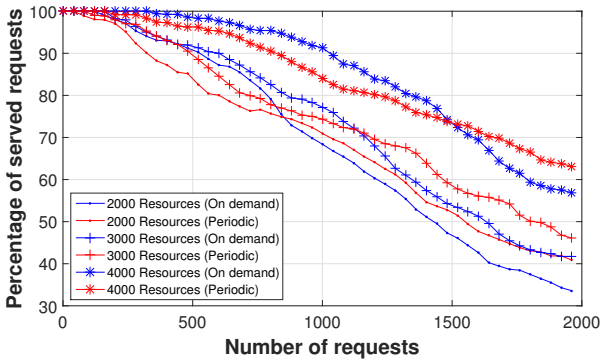

Fig. 2. Percentage of successful assignments.

TABLE II
TYPE OF FRIENDSHIP VALUES.

TYPE		F
Owner Object Relationship	OOR	1
Co-Location Object Relationship	CLOR	0.8
Co-Work Object Relationship	CWOR	0.8
Social Object Relationship	SOR	0.6
Parental Object Relationship	POR	0.5

level increases with the amount of available resources, since they allow to satisfy a higher number of application services. Moreover, we can notice how with low number of requests the on demand approach performs better than the periodic one, but when the number of request increases the periodic approach outperforms the other one. This phenomenon is due to the fact that the periodic approach is able to best allocate the available resources since the EN knows exactly all the requests that need to be satisfied, but on the other hand it has outdated information. The on demand approach enables the EN to allocate resources as soon as a request arrive, but not knowing the total number of requests that it will receive, it can only make sub-optimally choices.

The trustworthiness of the friends is calculated based on [11], where each device has different level of trust towards its friends according to the type of relations among them (see Table II).

Table III shows how the proposed algorithm reacts to different user preferences. Each user is characterized by a single preference assigned randomly among the three possible:

- maximize the trustworthiness of the application received;
- minimize the energy consumption from the user own devices;
- minimize the risk that the helping devices will leave the coverage area of the edge node.

Users with preference regarding the energy will try to match with plugged devices or with devices of other users in order to save their own resources. Finally, to minimize the failed tasks due to the devices' movements, user will prefer static or slow moving device.

TABLE III
EFFECTS OF DIFFERENT PREFERENCE SETTINGS ON THE MATCHING ALGORITHM

	Avg Trust of helping devices	Avg Consumed Energy by own devices	Failed Tasks due to movement
Max Trust	85%	67%	25%
Min Energy	48%	29%	34%
Min Task Fail	62%	46%	10%

Table III shows the effects of the three proposed preferences (in the rows), when the same user set a different preference for 15 different applications.

As expected, if a user wants to maximize her trustworthiness, the EN will select the devices tied by the OOR which are the most trustworthy, to the detriment of the consumed energy; on the other hand, to minimize the energy, the EN will choose other devices whatever the social relationship with the requester. The third preference is a good compromise since it is expected that the mobile devices of a user will likely be predictable in their movements, so the EN will choose the user' devices to minimize the failed tasks unless there is some static device that can give a higher reliability.

VI. CONCLUSION

In this paper, we proposed a novel federation mechanism, so called Social Mobile-IoT Clouds, in order to exploit the potential cooperation among co-location IoT devices by considering the social ties between them. Throughout this study, we have been focused on cooperative scenario, where, the edge node operates as an orchestrator to support dynamic resources sharing among devices. In order to manage the sharing resources, an assignment game using matching theory is presented, which the performance evaluation has shown the enhancement obtained in the term of successful assignments. For the future work we plan to develop the new trust algorithm and improve the matching solution. We can expand our point of view, and consider more than one network edge node, and users may change their positions constantly. Also considering matching in dynamic environment, where the preference of each device changed during the time, then the time is also must be considered as a dimension for the solution. In this work we assume that edge node is aware about resources and presence time of devices, we plan to design truthful mechanism that forces the devices reveal their duration of presence truthfully to the edge nodes.

ACKNOWLEDGEMENTS

This work was supported by Fondazione di Sardegna (Project ODIS, CUP: F72F16003170002).

REFERENCES

[1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[2] F. Bonomi, R. Milito, J. Zhu, and 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.

[3] N. Fernando, S. W. Loke, and W. Rahayu, "Mobile cloud computing: A survey," *Future generation computer systems*, vol. 29, no. 1, pp. 84–106, 2013.

[4] I. Farris, L. Militano, M. Nitti, L. Atzori, and A. Iera, "Mifaas: A mobile-iot-federation-as-a-service model for dynamic cooperation of iot cloud providers," *Future Generation Computer Systems*, vol. 70, pp. 126–137, 2017.

[5] E. Ahmed, I. Yaqoob, A. Gani, M. Imran, and M. Guizani, "Internet-of-things-based smart environments: state of the art, taxonomy, and open research challenges," *IEEE Wireless Communications*, vol. 23, no. 5, pp. 10–16, 2016.

[6] V. Pilloni, P. Navaratnam, S. Vural, L. Atzori, and R. Tafazolli, "Co-operative task assignment for distributed deployment of applications in wsns," in *2013 IEEE International Conference on Communications (ICC)*. IEEE, 2013, pp. 2229–2234.

[7] 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.

[8] S. Ranjbaran, A. Mohammadi, and M. H. Manshaei, "On cooperation mechanism in internet of things with multiple sponsors," in *Globecom Workshops (GC Wkshps), 2016 IEEE*. IEEE, 2016, pp. 1–6.

[9] 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.

[10] L. Wang, H. Wu, Z. Han, P. Zhang, and H. V. Poor, "Multi-hop cooperative caching in social iot using matching theory," *IEEE Transactions on Wireless Communications*, vol. 17, no. 4, pp. 2127–2145, 2018.

[11] M. Nitti, R. Girau, and L. Atzori, "Trustworthiness management in the social internet of things," *IEEE Transactions on knowledge and data engineering*, vol. 26, no. 5, pp. 1253–1266, 2014.

[12] C. Shi, V. Lakafosis, M. H. Ammar, and E. W. Zegura, "Serendipity: enabling remote computing among intermittently connected mobile devices," in *Proceedings of the thirteenth ACM international symposium on Mobile Ad Hoc Networking and Computing*. ACM, 2012, pp. 145–154.

[13] T. Nishio, R. Shinkuma, T. Takahashi, and N. B. Mandayam, "Service-oriented heterogeneous resource sharing for optimizing service latency in mobile cloud," in *Proceedings of the first international workshop on Mobile cloud computing & networking*. ACM, 2013, pp. 19–26.

[14] M. Nitti, V. Popescu, and M. Fadda, "Using an iot platform for trustworthy d2d communications in a real indoor environment," *IEEE Transactions on Network and Service Management*, 2018.

[15] I. Farris, R. Girau, L. Militano, M. Nitti, L. Atzori, A. Iera, and G. Morabito, "Social virtual objects in the edge cloud," *IEEE Cloud Computing*, vol. 2, no. 6, pp. 20–28, 2015.

[16] R. Girau, S. Martis, and L. Atzori, "Lysis: A platform for iot distributed applications over socially connected objects," *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 40–51, 2017.

[17] M. N. Soorki, M. H. Manshaei, B. Maham, and H. Saidi, "On uplink virtual mimo with device relaying cooperation enforcement in 5g networks," *IEEE Transactions on Mobile Computing*, vol. 17, no. 1, pp. 155–168, 2018.

[18] H. Zhu, D. Liu, S. Zhang, Y. Zhu, L. Teng, and S. Teng, "Solving the many to many assignment problem by improving the kuhn–munkres algorithm with backtracking," *Theoretical Computer Science*, vol. 618, pp. 30–41, 2016.

[19] C. Marche, L. Atzori, and M. Nitti, "A dataset for performance analysis of the social internet of things," 2018.