

MIFaaS: a Mobile-IoT-Federation-as-a-Service Model for Dynamic Cooperation of IoT Cloud Providers

I. Farris^a, L. Militano^{a,*}, M. Nitti^b, L. Atzori^b, A. Iera^a

^a*University Mediterranea of Reggio Calabria, DIIES Department, Italy*

^b*University of Cagliari, DIEE Department, Italy*

Abstract

In the Internet of Things (IoT) arena, a constant evolution is observed towards the deployment of integrated environments, wherein heterogeneous devices pool their capacities to match wide-ranging user requirements. Solutions for efficient and synergistic cooperation among objects are, therefore, required. This paper suggests a novel paradigm to support dynamic cooperation among private/public local clouds of IoT devices. Differently from *device-oriented* approaches typical of Mobile Cloud Computing, the proposed paradigm envisages an *IoT Cloud Provider (ICP) -oriented* cooperation, which allows all devices belonging to the same private/public owner to participate in the federation process. Expected result from dynamic federations among ICPs is a remarkable increase in the amount of service requests being satisfied. Different from the Fog Computing vision, the network edge provides only management support and supervision to the proposed Mobile-IoT-Federation-as-a-Service (MIFaaS), thus reducing the deployment cost of peripheral micro data centers. The paper proposes a coalition formation game to account for the interest of rational cooperative ICPs in their own payoff. A proof-of-concept performance evaluation confirms that obtained coalition structures not only guarantee the satisfaction of the players' requirements according to their utility function, but also these introduce significant benefits for the cooperating ICPs in terms of number of tasks being successfully assigned.

Keywords: IoT, Mobile Cloud Federation, Mobile Edge Computing, Fog Computing, Coalition Formation, Game Theory

*Corresponding author. University Mediterranea of Reggio Calabria, DIIES Department, 89122 Reggio Calabria, Italy. Email: leonardo.militano@unirc.it. Phone: +3909651693276

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

1. Introduction

The landscape of the Internet of Things (IoT) is being enriched by ever more heterogeneous devices and objects, which offer a wide range of capabilities depending on their specific hardware equipment.

Resource-constrained sensors and actuators, as well as high-performing nodes, such as smartphones and smart vehicles, offer indeed a wide range of services (e.g., computation, storage/caching, sensing/actuation). Therefore, actual implementation of highly-integrated Internet of Things services and added-value applications [1] may benefit from the design of methods for the effective cooperation among devices and network/cloud platforms in view of sharing information, services, and capabilities.

Over the past years, academic and industrial communities have proposed several paradigms to better exploit the potential of IoT devices [2]. Among the most promising ones is, undoubtedly, the deployment of Cloud platforms to support novel and advanced mobile applications, which are quickly and decisively impacting also on the design of IoT solutions. In the context of Mobile Cloud Computing (MCC) [3] [4] supporting task offloading and energy savings is the usual key objective. Instead, in the IoT domain [5] [6] the main goal is the provision of solutions that best exploit the huge amount of data and services offered by heterogeneous IoT devices.

A typical issue in traditional Cloud-based solutions, of undoubted concern for IoT pervasive environments, is the need to keep latency in data transmission to/from remote data centers under control. To reduce the impact of this parameter on the quality of the offered services, a recent popular approach is to move micro data centers towards the edge of the network. Mobile Edge Computing (MEC) [7], Cloudlets [8], [9], [10] and the recently emerging Fog Computing [11] paradigm in smart environments follow this trend. The cited solutions aim at creating a distributed infrastructure to effectively process data generated by billions of IoT devices.

A further source of performance enhancement is the opportunistic cooperation among mobile devices. The creation of dynamic local clouds, as proposed, e.g., in [12] and [13], can lead to a shorter service execution time and lower energy consumption. This result is a consequence of the increased computation, storage, and communication capabilities offered by a cooperative group of mobile devices. However, in IoT scenarios the opportunistic cooperation needs to include sensing, actuation, and location-based services other than computation, storage and networking cloud services, typical of mobile cloud computing settings. Besides, distributed approaches to task allocation usually introduce too stringent requirements into IoT devices and raise scalability and reliability issues in a highly dynamic environment.

On the basis of these considerations, this paper investigates on a novel hybrid paradigm, *Mobile-IoT-Federation-as-a-Service (MIFaaS)*, which exploits typical features of Fog computing to support dynamic cooperation among private/public clouds of IoT objects. The proposed solution allows for a better scalability and a significant reduction in complexity and costs in the network access

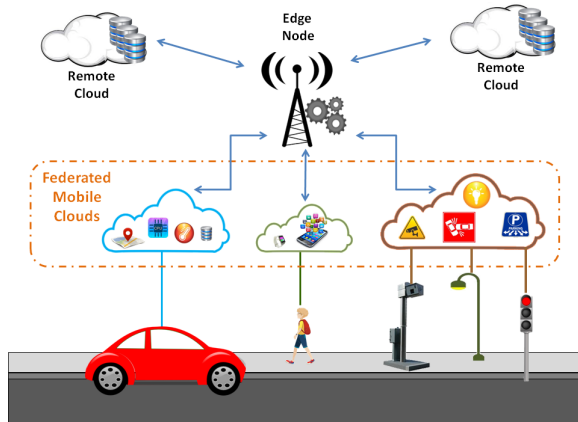


Figure 1: Mobile IoT cloud federation scenario.

segment compared to the deployment of micro data centers. In fact, according to the MIFaaS paradigm, objects themselves are able to provide resources “on-the-fly”. Nonetheless, different from the typical *device-oriented* approach in MCC solutions, where the single device can receive support from nearby opportunistic devices and their embedded resources only, in MIFaaS the attention is on the federation of pools of devices/objects managed by private/public owners, the so-called IoT Cloud Providers (ICPs). The proposed paradigm has the potential to introduce a broad set of advantages for end-users, network and service providers, well beyond those thoroughly investigated and validated in this paper; these advantages include: (i) enabling value-added services implemented through the federation of distributed IoT services and things belonging to different ICPs; (ii) costs savings derived from the reduced need to set up resource-hungry connections to the remote cloud, for service and management purposes; (iii) a better QoE, e.g., in terms of shorter latency in accessing a given service, by leveraging locally available resources (not only the ones provided by the owned devices); (iv) availability of a broader portfolio of services for end-users compared to the case in which they operate in a stand-alone manner (i.e., by leveraging only on their own resources).

The network edge node plays an important role in the implementation of the proposed paradigm for the reduction of the computational burden (and the energy consumption) in the local ICPs. As shown in Fig. 1, it hosts the functionalities to manage several tasks, such as: information collection about identity and capabilities of the devices; resource abstraction and definition of a common access interface to the services offered by local IoT devices (based on virtualization techniques); collection of service requests and, above all, dynamic management of cloud federations according to device capabilities and service requirements. The edge node hosts the “orchestrator”, which manages federations among public/private IoT clouds and enables the efficient sharing of their resources to deliver highly-integrated IoT services, whereas task execution is

completely delegated to IoT devices. The objective is to support ICPs in taking the best decisions regarding their federation by considering them as *rational self-interested entities* that aim at maximizing their own benefits (i.e., their own *utility*). The role of “orchestrator” may be played by either the Telco operator or by any other third-party entity who has a business interest in offering additional and enhanced services to its subscribers. This is the case, for example, of a coffee shop owner, a university, a theatre, or a smart city municipality.

The major contributions of this paper are:

- The proposal of a novel Mobile-IoT-Federation-as-a-Service (MIFaaS) solution to support the federation of private/public mobile IoT clouds in heterogeneous IoT environments;
- The definition of a coalition formation game model allowing the rational and self-interested ICPs to take their grouping choices and converge to a *Nash-stable* solution in which no ICP has incentive to leave the proposed solution;
- A proof-of-concept performance evaluation campaign to demonstrate the enhancements introduced by the proposed solution w.r.t. classic device-oriented solutions in terms of number of task requests being successfully allocated and in terms of ICP satisfaction.

The paper is organized as follows. Section 2 provides an overview of the related works. In Section 3 we describe the proposed system and reference scenario for the edge-assisted ICPs federation. Section 4 and Section 5 describe the game theoretic coalition formation model and the conducted performance evaluation campaign, respectively, whereas a discussion on future works and open challenges is presented in Section 6. Final remarks are given in the concluding Section.

2. Related Work

2.1. Mobile Cloud Computing and Fog Computing

The scientific literature on Mobile Cloud Computing [14] has extensively investigated the interaction among mobile devices and a fixed data center to provide resources to deliver the desired service. In particular, authors of MCC [15] [16] have investigated optimization solutions for the application design and deployment, whereas dynamic scheduling policies for adapting task offloading to computation workload, time constraints, and network connectivity are the subjects of articles such as [17] [18]. Cooperative Mobile Cloud schemes can be straightforwardly classified into two schemes. Infrastructure-less cooperative solutions to create local mobile clouds have gathered a wide interest [12], [13], [19]. In particular, these solutions rely on a distributed cooperation to create collaborative computing platforms where the devices mutually share their resources. The main drawback of such a distributed approach is the signaling overhead needed to coordinate the cooperation activities and to build an updated and

shared knowledge of the current network state, the available resources, and the received service requests. On the other hand, infrastructure-based approaches foresee a task offloading from mobile devices to remote high-capability data centers, as proposed, for instance, in [3] and [4]. Typical limitations of the approach are in terms of latency and bandwidth requirements for communication with a remote server. To overcome these drawbacks, the authors of [8] proposed the concept of micro data centers, deployed in the network access points and known as Cloudlet. The extension of the Cloudlet concept into the IoT scenario, resulting in the definition of the so-called Fog Computing [20], brings along with it the necessity of solving several specific IoT-related challenges, such as for example the management of a huge number of heterogeneous IoT devices involved in the process. To this aim, there is a growing impetus in the design of cloudlet-based solutions able to effectively support sensing and actuation applications [21] [22]. Telco providers are facing a trade-off between the correct dimensioning of resources to support IoT services and the financial costs deriving from the deployment of a large number of processing nodes located very close to the customers. The MIFaaS paradigm proposed in this paper supports Telco providers in achieving the above mentioned trade-off by leaving control and orchestration functions to the edge nodes (thus reducing their processing capabilities), and completely delegating task execution to IoT devices. The idea of a resource coordinator elected among the mobile nodes is proposed in [23] to manage the matching between application requests and device resources. The authors of [24] consider IoT Cloud federations for physically dislocated platforms with the focus on authentication and security issues, whereas in [25] the authors propose the refactoring of the Cloudlet in a controller so that nearby devices can be configured into a coordinated cloud computing service. However, both papers do not actually tackle the heterogeneity of IoT objects in terms of capabilities and resource constraints. As a further difference with our research, a single mobile local cloud is considered, thus suffering from scalability and flexibility issues.

2.2. Game theoretic models for federation formation

A preliminary analysis of edge-assisted collaboration of local clouds has been proposed in [26]. In this paper, we take forward our research by proposing a game theoretic model for the federation formation problem, which allows to better capture the *rationality* of the users (the ICP owners) and their willingness to join or leave a federation according to their personal preferences over the utilities they reach. We have also extended the analysis in a wider set of scenarios and proved not only that a high number of ICPs increases the percentage of satisfied tasks, but also that our federation approach is able to guarantee the ICP satisfaction. A current open challenge to promote the effective deployment of federations among private/public ICPs is to identify adequate sources of motivation and incentives that facilitate the end-user decisions to share their private resources within the federation. If the framework is able to guarantee the desired ICPs' outcomes according to their specific preferences, then ICPs will be more willing to adopt the proposed solution. Therefore, the game theoretic modelling and the relevant stability analysis plays a fundamental role to

reach this goal. Indeed, several researchers [27] [28] have recently applied game theoretic solutions to model the interaction among devices and to share the increased revenues deriving from a cooperative resource sharing in the context of mobile cloud computing. In particular, cooperation among mobile cloud service providers is investigated in [27], where available resources are shared and better utilized to increase revenues. Classic game theoretic solutions model the way to share the increased revenues. Differently, the authors of [28] propose a congestion game to model a power management scheme for mobile cloud computing to minimize power consumption of both the servers on the cloud and the mobile devices. Our work differentiates itself from the cited papers as it addresses methods to enable effective federations of local clouds. In this area as well, several research activities targeted to study cooperative approaches among cloud providers have been conducted. Coalitional game theory concepts have also been considered in [29] to formulate an economic model for the federation of virtualized computing and network resources, where the Shapley value is proposed as a means for participants to share the value of a federation. Close to our problem, the authors of [30] and [31] study a coalition formation game for a set of Cloud providers willing to federate to increase their revenues. In particular, [30] proposes a distributed algorithm to cooperatively set up federations that allow to increase the individual benefit in terms of energy cost reduction with respect to the case in which users work in isolation. While [31], instead, addresses a method to form federations of cloud providers so to maximize their profit when a set of resources is requested. A key difference with the model for the ICPs federation in this paper is in the need to consider the extreme heterogeneity of IoT services and resources of the devices belonging to ICPs. To this aim, the key features of the model, such as the utility function and the preference relation for the players in the game, are tailored to the specific problem and challenges of ICP federation. With regard to the transferable utility game theoretic approach used in [30] and [31], a further difference is that the utility for the proposed game is non-transferable and is defined by a weighted function of the number of tasks executed for a player and the resources used by the same player in a cooperative federation of personal clouds.

2.3. Task allocation in the IoT

An additional issue our paper addresses is task allocation for the IoT. Although task mapping has been extensively analyzed in the literature [32], IoT introduces new constraints and challenges that require an appropriate re-formulation of the problem. Indeed, IoT applications typically require the composition of very different tasks. In a smart city environment, for example, a common application request is to manage a traffic jam. The needed tasks are related to the *sensing of the environment*, such as counting the number of cars involved and the traffic flow in other areas of the city, the *real-time processing* of the information obtained, and the *activation of physical objects*, e.g., traffic lights, to address the traffic jam. IoT solutions need to manage the interdependencies of these complex IoT services, and to provide efficient methods to allocate the relevant elementary tasks to the available IoT devices, while verifying the

satisfaction of the application constraints. To this aim, the literature [33] [34] provides several interesting heuristics. In particular, the authors in [33] model IoT services as task graphs in terms of data streams and operations to be distributed over the network; a task mapping algorithm is available, which aims at minimizing the consumption of resources among the available nodes. In [34] instead, group of nodes able to perform similar tasks cooperate to distribute the relevant requests by leveraging a consensus-based approach. In the context of IoT Cloud platform federation, the VITAL project proposes an architecture of Cloud of Things [35] by defining interface virtualization as the enabling means to access data and services from heterogeneous objects supported by different IoT ecosystems. These activities do not focus on task allocation and federation algorithms; thus can be considered as complementary to our study in the view of providing users with a global IoT ecosystem.

3. Reference Scenario and System Model

In the reference scenario, we define an ICP as a local cloud of connected devices belonging to the same owner. The ICP owner may be a private user, a public authority (e.g., a Smart City municipality), a service provider, the owner of a Wireless Sensor Network (WSN), the owner of a power network, among others. In general, devices belonging to a single ICP may differ a lot from one another. For instance, a private ICP owner may register devices such as a smartphone, a laptop or a smartwatch, whereas a public ICP owner (Smart City municipality) may register roadside units, pollution sensors, smart traffic lights, security cameras, and others. As already discussed, the orchestrator in the edge node is responsible for the creation and maintenance of opportunistic federations to enable the provision of advanced IoT services. In our reference scenario, we assume that the edge node is the first network access point for the ICP. In particular, we consider small area scenarios, such as a city neighborhood, a shopping mall, a theatre, a stadium, a university campus. Section 6 presents a discussion on possible research directions to extend the proposed solution to wide-area scenarios, where multiple edge nodes are taken into account. Fig. 1 shows a reference scenario adhering to the MIFaaS philosophy and illustrating the presence of different private/public ICPs.

When the ICP owner registers its devices to the reference edge node, then a virtual counterpart for the ICP is created, the so-called “ICP controller”. This is a software component running in the edge node and responsible for the resources and services management of the corresponding ICP. To this aim, the ICP controller maintains an updated status of the services offered by the devices forming the ICP and of their available resources (e.g., computation, processing, and sensing).

The ICP controller is able to address the heterogeneity of the physical objects, since all the communications among objects are handled at a digital level, where the ICP controllers of the same Telco provider interoperate through the use of the same semantic description.

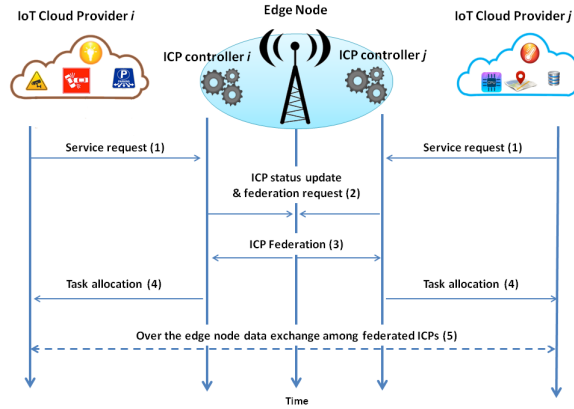


Figure 2: Coordination flow.

Whenever a new request of resources comes from an application running on a device of the ICP, the ICP controller is informed and handles it. A first attempt is made to handle the request by only exploiting the devices belonging to the ICP. In case the devices under its control are not sufficient to implement the task, then a collaboration request is triggered to other ICP controllers running in the same edge node. In this case, the orchestrator in the edge node acts as a broker, which receives requests and status updates from the ICP controllers and implements a federation algorithm to cooperatively offer added-value IoT services by pooling together the capabilities and resources of the involved ICPs.

Fig. 2 highlights the procedure to implement the proposed solution. The process starts with an application running on a user device that sends a service request to the ICP controller at the edge node (1). If the set of devices belonging to the ICP has not enough capabilities to handle the services, then a federation request is sent to the edge node acting as *broker* or orchestrator of the incoming requests. A software module at the edge node handles the request, which is received together with information about the ICP status (2). After the collection of a set of resource requests over a predetermined time interval, the edge node triggers the federation formation process aimed at sharing the resources in terms of computational, communication, storage, and information. The edge node performs the resource management by assigning each task to the most fitting object in the federation. Once the edge node decides the federation, it notifies the “requesting ICP controllers” about the “supporting ICPs”, and specifies the number and type of tasks to be offloaded (3). Then the ICP controller transfers to the specific devices the information about the tasks they have to perform (4). Finally, the federated ICPs exchange input and output data through over-the-edge communications (5).

It clearly emerges the importance of establishing a mechanism to wisely allocate the tasks and handle the mobility of the ICPs. We address this issue in the remainder of the paper. It is worth highlighting that the orchestrator in the

edge node only supports the creation and management of the federation among ICPs, by mediating the received requests and facilitating the synchronization of the resource status for each ICPs' device. It does not provide the final services to users. Limiting the edge node role only to management responsibilities reduces the costs of its deployment. This is a key distinctive feature of the MIFaaS approach with respect to other solutions, such as Fog computing for example, where the peripheral network nodes provide additional computing resources.

3.1. Service Model for the Edge-assisted ICP Federation

Without losing in generality, we consider a scenario with a single edge node and several ICPs. The main objective of the edge node activity is to create and orchestrate one or more federations of ICPs so that the number of executed tasks is maximized, under the constraints set by the rational and self-interested ICPs.

We consider a network with a set \mathcal{N} of ICPs, where the i -th ICP is endowed with a set of \mathcal{D}^i devices. Resources can belong to a set of \mathcal{M} different resource types. Each resource of the m -th type associated to the i -th ICP (i.e., offered by one of its devices) belongs to a set $\mathcal{R}^{i,m}$. In the remainder of the paper, the generic r -th resource is identified by $res_{i,m,r}$. The i -th ICP has a workload of application requests mapped at the edge node onto task requests. The t -th task request of the m -th type is identified by $task_{i,m,t}$ and belongs to the set of task requests denoted by $\mathcal{T}^{i,m}$. Note that task request types and available resource types coincide and are thus identified by the same set of types denoted by \mathcal{M} .

A further important feature characterizing an ICP is its mobility. An ICP is a local cloud of heterogeneous objects; we assume that all devices belonging to the same ICP have the same mobility pattern. An example may be a pedestrian user carrying multiple devices, or a car driver in a vehicle with multiple devices around him. In particular, we make the following distinction: an ICP can be either (i) *static*, like roadside units or a public traffic light monitoring unit, or (ii) *mobile* at a low speed, like a private person moving around the city (*pedestrian* mobility profile), or (iii) mobile at a higher speed, like a vehicle (*vehicular* mobility profile). The mobility parameter is of high concern during task allocation, as it influences the probability that a task is performed and the result is timely received by the requesting ICP. As mentioned in the previous section, we assume an "edge node assisted" solution, whereby once a task is allocated to an ICP, the data exchange relevant to that task goes over the edge node (solution also proposed in [25]). Therefore, it is important that both requesting and executing ICPs remain under the coverage of the edge node and maintain the connection to the corresponding ICP controllers, for the time required to execute the task and transfer the results. We model this by foreseeing that the edge node is able to predict the mobility pattern of the ICPs, so that it can get a rough estimation of the probability that an ICP leaves the area of interest and thus loses the contact with its ICP controller. To this aim, in our model we include a mobility factor $p_i \in [0, 1]$ for the i -th ICP, which accounts for the contact time probability between an ICP and the serving edge node: 0 stands for a static ICP, while 1 for an ICP moving away before the

task conclusion with probability one. The research community is active on the issue of user mobility pattern prediction in mobile and wireless environments, with focus on the modelling of key parameters such as contact time duration and the number of contact times [36] also based on the analysis of real traces for human mobility [37]. We rely on these research contributions to support the assumption that the edge node has the possibility to estimate the contact time duration with the ICPs and by this include this knowledge in the federation formation process. We are aware that in IoT scenarios other communication related aspects may influence the probability that devices belonging to different ICPs can actually exchange data. Notwithstanding, the scope of our paper is not to provide any novel contribution in the characterization of ICP mobility, but to evaluate the impact the ICP mobility has on the overall federation process and its performance.

In our problem formulation, the requesting ICP and the ICP which implements the task act in a federation. The event that the edge node loses the contact with any of them should be avoided before the task output is successfully delivered to the requesting application. The case in which this event is avoided is called *successful task cooperation*. Let p_i represent the probability for i -th ICP to leave the edge node coverage before the task cooperation ends, then the successful task cooperation probability can be written as follows

$$P_j(task_{i,m,t}) = P(A) \cap P(B) = P(A)P(B|A) \quad (1)$$

where $task_{i,m,t}$ is requested by the i -th ICP and executed by the j -th ICP, event A is the probability of i -th ICP remaining under the edge node coverage ($1 - p_i$), and event B is the probability j -th ICP remaining under the edge node coverage ($1 - p_j$). We also introduce a *compromise factor* $\alpha_i \in \{0, 1\}$ that gives a measure of the willingness of the i -th ICP to make its own resources available to the rest of the community so that other federated ICPs may use them for their tasks. Finally, accounting for the different features of the IoT services, we classify the tasks into *exclusive-use tasks* and *shared-use tasks*. The former are those whose output can be used by a single ICP only; such as the case of computational tasks where the relevant physical resources are typically allocated to the specific needs of a single ICP. The latter are those whose result is usually useful to multiple ICP requests, such as the case of sensing activities whose output can be reused by multiple services. Indeed, sensing applications usually require measurements of a physical phenomenon, relevant to a specific location and with a desired level of Quality of Information (QoI). Even if multiple QoI metrics could be defined in terms of precision, timeliness, completeness, and relevance [38], in our paper we focus on data completeness which requires that several measures are taken from different sensor devices to provide an accurate and reliable sensing estimation. These measurements can be shared among multiple applications if they are relevant to the same phenomenon, location and time. A practical example is the case where two applications of different ICPs require the current temperature information in the same location. Instead of using two different sensing devices to collect the same information, it is more

Table 1: Notations and symbols.

\mathcal{N}	Set of ICPs
\mathcal{D}^i	Set of devices for the i -th ICP
\mathcal{M}	Set of resource types
$\mathcal{R}^{i,m}$	Set of resources of the m -th type at the i -th ICP
$\mathcal{T}^{i,m}$	Set of task requests of the m -th type from the i -th ICP
$task_{i,m,t}$	t -th task request of the m -th type from the i -th ICP
$res_{i,m,r}$	r -th resource of m -th type available at the i -th ICP
α_i	Compromise factor for the i -th ICPs
$\mu_{i,m,t}$	Reuse factor for the t -th task request
p_i	Mobility factor for the i -th ICP
$P_j(task_{i,m,t})$	Success probability for task $task_{i,m,t}$ being executed by the j -th ICP
$\mathcal{TQ}(\mathcal{S})$	Set of requested tasks in federation \mathcal{S}

efficient to share the information among the interested ICPs. This justifies the introduction of the so-called *reuse factor* associated to a task, namely $\mu_{i,m,t} \in \{0, 1\}$, which represents the measure of the possibility for a task to be reused by multiple applications (e.g., a reuse factor of 1 may be set for sensing data). Please refer to Table 1 for the list of notations used in the paper.

4. A Coalition Formation Game for the ICP Federation

To model the cooperation problem among the ICPs we use the coalitional game theory [39]. We define a non-transferable utility (NTU in short) coalitional game $(\mathcal{N}, \mathcal{V})$, where \mathcal{N} is a set of N players and \mathcal{V} is a function, such that, for every coalition $\mathcal{S} \subseteq \mathcal{N}$, $\mathcal{V}(\mathcal{S})$ is a closed convex subset of $\mathbb{R}^{|\mathcal{S}|}$. The latter contains the payoff vectors that the players in \mathcal{S} can achieve, and $|\mathcal{S}|$ is the number of members in the coalition \mathcal{S} . The game is in *characteristic form* because the achievable utility in a coalition only depends on the players forming the coalition and not on the other players in the network. We define $\mathcal{V} : \mathcal{S} \rightarrow \mathbb{R}^{|\mathcal{S}|}$ such that $\mathcal{V}(\emptyset) = \emptyset$, and for any coalition $\mathcal{S} \subseteq \mathcal{N} \neq \emptyset$ it is a singleton set $\mathcal{V}(\mathcal{S}) = \{\mathbf{v}(\mathcal{S}) \in \mathbb{R}^{|\mathcal{S}|}\}$ where each element of the vector $\mathbf{v}(\mathcal{S})$ is the value $v_i(\mathcal{S})$ associated to each player $i \in \mathcal{S}$.

The objective for the players in this NTU game is to maximize their associated value in the coalition they belong to. Such a value is considered as the difference between the gain that the players obtain (the utility) in cooperation minus a cost term associated to the sharing of their resources. The utility term for any ICP is associated to the number of tasks being executed over the number of tasks requested, whereas the cost term is a measure of the amount of resources used to execute all assigned tasks over the total amount of available resources at the ICP. Moreover, we introduce a weighting term in the definition of the player value, the so-called *compromise factor*, which measures the importance attributed to the cost the player has to face to obtain the utility. This parameter allows to better model the influence of the ICPs rationality and explicitly consider their willingness to make their own resources available to the rest of the community, so that other federated ICPs may use them for their

own tasks. This reflects a subjective interpretation of utility and costs for each player. In particular, its value may change over time according to the current ICP conditions in terms of available resources and importance/priority of the tasks. For instance, in case of high priority services, the ICP is more interested in achieving her goal rather than saving her own resources leading to a low value selection for the compromise factor. Differently, when the ICP is willing to keep the usage of the own resources low (e.g., due for example to the scarcity of resources), then it will set the compromise factor to a high value so that it will be involved only in highly advantageous federations. To simplify the analysis of the results and without losing generality, during the performance evaluation (see Section 5) we consider all players having a constant and equal compromise factor.

It is evident that having a cost term in the characteristic function, leads to a non-superadditive game with disjoint coalitions forming a partition $\Pi \in \Pi(N)$, where $\Pi(N)$ is the set of all partitions. In general, the set of all possible partitions of \mathcal{N} has a total number of B_N , where B_N is the N -th Bell number [40], and it grows exponentially with the number of players N . Thus, finding the optimal partition via exhaustive search through all possible partitions, being an NP-complete problem [41], is unfeasible. An alternative solution is to introduce a preference relation over coalitions so that the game is written in the form of an *hedonic game* [42] and players adopt switch operations based on their preferences over the potential coalitions to be a member of, based on who else is in the coalition. We thus define a *collection* of coalitions \mathcal{C} as the set $\mathcal{C} = \{\mathcal{C}_1, \dots, \mathcal{C}_l\}$ of mutually disjoint coalitions $\mathcal{C}_i \subset \mathcal{N}$ such that $\mathcal{C}_i \cap \mathcal{C}_{i'} = \emptyset$ for $i \neq i'$. If the collection contains all players in \mathcal{N} , i.e., $\bigcup_{i=1}^l \mathcal{C}_i = \mathcal{N}$, then the collection is a *partition* Π or *coalition structure* [39].

In our problem, for a given federation $\mathcal{S} \subseteq \mathcal{N}$ we determine the value for a player i (i.e., ICP i) in the federation as a function of its utility term $u_i(\mathcal{S})$ and its cost term $c_i(\mathcal{S})$. In particular the utility term $u_i(\mathcal{S})$ is a measure of the number of tasks being *successfully executed* over the number of requested tasks, which is defined as:

$$u_i(\mathcal{S}) = \frac{\sum_{m \in \mathcal{M}} \sum_{t \in \mathcal{T}^{i,m}} \sum_{j \in \mathcal{S}} x_{i,m,t,j} \cdot P_j(task_{i,m,t})}{\sum_{m \in \mathcal{M}} |\mathcal{T}^{i,m}|} \quad (2)$$

where $x_{i,m,t,j}$ is a binary variable assuming value 1 if task $task_{i,m,t}$ requested by the i -th ICP is implemented by the j -th ICP (note that i may be equal to j if task $task_{i,m,t}$ is implemented by the requesting ICP itself) and value 0 otherwise, whereas $P_j(task_{i,m,t})$ is the successful task cooperation probability defined in (1) and $\mathcal{T}^{i,m}$ is the set of task requests for the i -th ICP of type m .

The cost term, instead, is a measure of the available resources of any single ICP that are being used in the given coalition. To account for the heterogeneity of the ICPs, we define different weights as it follows. Given the i -th ICP, we introduce a weighting factor w_i as a function of the number of different resource

types at the i -th ICP, that gives the corresponding weight to the resources being used to solve a task. Thus, the cost term for the i -th ICP in a federation $\mathcal{S} \subseteq \mathcal{N}$ is defined as:

$$c_i(\mathcal{S}) = \sum_{m \in \mathcal{M}} w_i \frac{\sum_{j \in \mathcal{S}} \sum_{t \in \mathcal{T}^{j,m}} y_{j,m,t,i} \cdot \text{task}_{j,m,t}}{\sum_{r \in \mathcal{R}^{i,m}} \text{res}_{i,m,r}} \quad (3)$$

where $y_{j,m,t,i}$ is a binary variable assuming value 1 if task $\text{task}_{j,m,t}$ requested by the j -th ICP is implemented by the i -th ICP (j may be equal to i if the performed task $\text{task}_{j,m,t}$ is requested by the executing ICP itself) and value 0 otherwise. Noteworthy, in those cases where $\mathcal{R}^{i,m} = \emptyset$, the corresponding m is not considered in the summation to prevent the cost from raising to an infinite value.

We are now ready to define the value for the single i -th ICP in a federation $\mathcal{S} \subseteq \mathcal{N}$ as:

$$v_i(\mathcal{S}) = \begin{cases} -\infty & \bigcup_{m \in \mathcal{M}} \mathcal{T}^{i,m} = \emptyset \\ u_i(\mathcal{S}) - \alpha_i \cdot c_i(\mathcal{S}) & \text{otherwise} \end{cases} \quad (4)$$

where α_i is the so-called *compromise factor*. In eq. (4), the first condition refers to the case where a player has no task requests and we prevent this player from joining any coalition. The rationale behind this choice is that a player is self-interested and *non-altruistic* in the sense that it is not interested in offering his resources (cost) without obtaining any utility for it (task requests). Noteworthy, this condition may be removed in case other design solutions are preferred. Finally, the value for a generic federation $\mathcal{S} \subseteq \mathcal{N}$ of ICPs is defined as:

$$v(\mathcal{S}) = \begin{cases} 0 & \mathcal{TQ}(\mathcal{S}) = \emptyset \\ \sum_{i \in \mathcal{S}} v_i(\mathcal{S}) & \text{otherwise} \end{cases} \quad (5)$$

where $\mathcal{TQ}(\mathcal{S}) = \bigcup_{i \in \mathcal{S}} \bigcup_{m \in \mathcal{M}} \mathcal{T}^{i,m}$ is the set of task requests in the federation. Note that, the first option is representative of the situation where no tasks are resolved in the considered coalition, in which case the value is readily equal to zero.

To compute all the terms of this model, a *task allocation algorithm* is required to map the set of task requests in the federation with the available resources. Task allocation is used by the edge node to define the value for the players and the coalition. If the coalition is part of the final partition, then the proposed task allocation will be actually implemented by the cooperating ICPs. Being task allocation not the main focus of this paper, we consider a simple algorithmic solution, based on a greedy approach. More sophisticated and effective algorithms can be easily taken from the literature and used without affecting the validity and the generality of the proposed cooperative paradigm. In particular, the chosen task allocation algorithm aims at mapping as many task requests as possible to the resources of the involved ICPs, compatibly with

the resources available at each device. When a new task from the list of total requested tasks must be allocated, the edge node performs the following steps: (i) if the reuse factor $\mu_{i,m,t}$ for the task is equal to one, then the edge node looks among the tasks already allocated to verify whether any of the corresponding allocated resources can also serve the new considered request; otherwise it (ii) starts searching for a suitable device with the needed resources by first ranking the available nodes in terms of mobility pattern and number of currently serving tasks; (iii) then it selects the device with the lowest mobility factor and number of executed tasks, so that higher probability of task execution success and load balancing are enabled; (iv) it allocates the task to the corresponding ICP owning that device; and eventually, (v) the ICP controllers update the resource availability status relevant to all the involved devices.

4.1. Coalition formation algorithm

The non-superadditivity of the cooperation model entails the formation of disjoint coalitions leading to a coalition formation game. A key approach to coalition formation is to enable players to join or leave a coalition based on well-defined preferences. The *preference order* \succ_i for any player $i \in \mathcal{N}$, is defined as a complete, reflexive, and transitive binary relation over the set of all coalitions that player i can possibly form, i.e., the set Π_i of coalitions containing i . In particular, for each player i , if $\mathcal{C} \succ_i \mathcal{C}'$, then this means that i prefers being a member of coalition \mathcal{C} more than coalition \mathcal{C}' . A less restrictive preference order is $\mathcal{C} \succeq_i \mathcal{C}'$, whereby player i prefers coalition \mathcal{C} at most as much as coalition \mathcal{C}' . The criterion or preference order to be used for comparing two partitions can either be coalition payoff orders or individual payoff orders. In this paper the preference order is defined so that, for each ICP $i \in \mathcal{N}$ and for all $\{\mathcal{C}, \mathcal{C}'\} \in \Pi_i$, we say that:

$$\mathcal{C} \succ_i \mathcal{C}' \Leftrightarrow f_i(\mathcal{C}) > f_i(\mathcal{C}') \quad (6)$$

where the preference function $f_i(\cdot)$ is defined for any ICP i and any coalition \mathcal{C} such that:

$$f_i(\mathcal{C}) = \begin{cases} v_i(\mathcal{C}) & \mathcal{C} \notin H(i) \text{ or } (|\mathcal{C}| = 1) \\ -\infty & \text{otherwise} \end{cases} \quad (7)$$

In particular, $v_i(\mathcal{C})$ is the payoff for ICP i in coalition \mathcal{C} , which is defined as in Eq. (4); $H(i)$ stores the history of the coalitions already visited during the algorithm execution to avoid an ICP to visit twice the same coalition that was left based on the preference of one of the players in the coalition, with the exception of the singleton coalitions (i.e., the coalitions with cardinality equal to one) [43]. In other words, any ICP i prefers being a member of coalition \mathcal{C} over \mathcal{C}' if its individual payoff is increased.

The so-defined preference order is at the basis of the *switch operation* for the coalition formation. In particular, given a partition $\Pi = \{\mathcal{C}_1, \dots, \mathcal{C}_l\}$ on the set of players \mathcal{N} and the preference relation defined in equation (6), an ICP $i \in \mathcal{N}$ decides to switch coalition (and consequently modifies the coalition

structure), leaving its current coalition $\mathcal{C}_\Pi(i)$ to join another different coalition $\mathcal{C}_k \in \Pi \setminus \mathcal{C}_\Pi(i) \cup \emptyset$ if and only if it prefers being part of the new coalition according to the preference relation, i.e., $\mathcal{C}_k \cup \{i\} \succ_i \mathcal{C}_\Pi(i)$. When a switch operation is applied by a player i , the history of the visited coalitions $H(i)$ is updated so that also $\mathcal{C}_\Pi(i)$ is included. Similarly, the current partition Π is to be updated as follows:

$$\Pi' = (\Pi \setminus \{\mathcal{C}_\Pi(i), \mathcal{C}_k\}) \cup \{\mathcal{C}_\Pi(i) \setminus \{i\}, \mathcal{C}_k \cup \{i\}\}.$$

The coalition formation game for the cooperative ICPs can be implemented as reported in Algorithm (1). Starting from an initial partition $\Pi^i = \mathcal{N} = \{\{1\}, \{2\}, \dots, \{N\}\}$, the edge node iteratively applies the switch operations until it converges to a final partition Π^f where no further switch operations are applied.

Data: Set of ICPs \mathcal{N}

Result: Coalition structure Π^f

Partition the network in $\Pi^i = \mathcal{N} = \{\{1\}, \{2\}, \dots, \{N\}\}$;

Initialize history function $H(i) = \emptyset, \forall i \in \mathcal{N}$;

Set the current partition as $\Pi^c = \{\mathcal{C}_1, \dots, \mathcal{C}_l\} = \Pi^i$;

repeat

for all $i \in \mathcal{N}$ **do**

for all $\mathcal{C}_k \in \Pi^c \cup \emptyset$ **do**

if $(\mathcal{C}_k \cup \{i\} \succ_i \mathcal{C}_{\Pi^c(i)})$ **then**

$\Pi' = (\Pi^c \setminus \{\mathcal{C}_{\Pi^c(i)}, \mathcal{C}_k\}) \cup \{\mathcal{C}_{\Pi^c(i)} \setminus \{i\}, \mathcal{C}_k \cup \{i\}\}$

$\Pi^c = \Pi'$

$H(i) = H(i) \cup \mathcal{C}_{\Pi^c(i)}$

end

end

end

until Converge to final Nash-stable partition $\Pi^f = \Pi^c$;

Algorithm 1: Coalition formation for cooperative ICPs

4.2. Stability properties for the ICP coalition formation game

The stability property of coalition structures can be studied through the Nash-stability concept, see [44], [45] and [42].

Definition 1 (Nash-stable partition). A partition $\Pi = \{\mathcal{C}_1, \dots, \mathcal{C}_l\}$ is Nash-stable if $\forall i \in \mathcal{N}, \mathcal{C}_\Pi(i) \succ_i \mathcal{C}_k \cup \{i\}$ for all $\mathcal{C}_k \in \Pi \setminus \mathcal{C}_\Pi(i) \cup \emptyset$.

In words, a partition Π is said Nash-stable if no player i has an incentive to move from its current coalition to join a different coalition of Π or to act in a non-cooperative way. Given the finite number of possible switch operations that can be applied, starting from the initial coalition structure Π^i , it can be proven that the proposed algorithm (1) always converges to a final partition Π^f of disjoint coalitions [46] [30]. In fact, based on the switch rule after each

switch operation, the resulting coalition is preferred to the previous coalitions. Consequently, a tested feasible coalition structure cannot be created again by any further switch operation. Since the total number of partitions is finite, the switch iterations will end by converging to a final partition.

Proposition 1. *Any final partition Π^f for the coalition formation algorithm in Algorithm 1 is a Nash-stable partition [42].*

PROOF. This proposition can be proved by contradiction, assuming that the final partition Π^f is not Nash-stable. This means that there exist a player $i \in \mathcal{N}$ and a coalition $\mathcal{C}_k \in \Pi^f \cup \emptyset$ such that $\mathcal{C}_k \cup \{i\} \succ_i \mathcal{C}_{\Pi^f}(i)$. If this is the case, player i will perform a switch operation which contradicts the assumption that Π^f is the final partition of the algorithm.

Nash-stability captures stability notions by only focusing on the movement of single players. Other stability concepts, such as the *core-stability*, model the stability considering the movements of groups of players; anyway, these are computationally hard to obtain. Therefore, Nash-stability is usually considered as a good trade-off and our algorithm always converges to a partition Π^f that is Nash-stable.

5. Performance Evaluation

In this section we evaluate the performance of the proposed MIFaaS paradigm by considering heterogeneous IoT scenarios where ICPs can request services, whose elementary tasks include computation, storage, and different types of sensing activities. The adopted tool is an ad-hoc conceived simulation environment implemented in Matlab. In particular, our simulator analyses the federation process as a time-slotted solution, by matching ICP service requirements with the available device resources. By this we mean that after collecting a set of requests over a predetermined time interval, the edge node evaluates the cooperation strategies among the ICPs. At this time, our simulator analyses the federation process, by matching ICP service requirements with the available resources and iterating the coalition formation process until convergence to a final stable coalition structure of federated ICPs. As an alternative, a discrete-event simulator could be implemented to better model the evolution over time of the scenarios. The simulation environment can be further enhanced by implementing a detailed physical layer. Nonetheless, the federation operations are run in the edge node based on the virtual counterparts of the physical objects. As a consequence, once the physical devices are matched into the virtual space, they become independent of their physical characteristics.

The performance evaluation aims at: (i) presenting results relevant to the *percentage of served tasks* and the *resources per ICP used* when adopting the proposed solution compared to alternative solutions, and (ii) giving insights on the proposed game theoretic solution for the federation formation in terms of *number of federations* formed, and *percentage of satisfied ICPs*. In our study, a

Table 2: Main simulation parameters

Parameter	Value
Number of ICPs	[4-22]
Number of devices per ICP	3
Computation units per task request	[1-5] MFLOPS
Storage units per task request	[10-50] MBytes
Number of sensing types	4
Sensing units per task request	[1-5]
Computation capacity per device	[1-10] MFLOPS
Storage resource units per device	[10-100] MBytes
Sensing resource units per device	[0-1]
Number of runs per simulation	500
Mobility factor p_i	[0.2]
Compromise factor α_i	[0-1]

self-interested ICP is considered as satisfied if it obtains a utility (see definition in (4)) that is larger or equal to the utility it receives when working alone without joining a cooperative federation. A wide range of simulative scenarios are considered where the values of the main parameters of the proposed model are suitably tuned for IoT environments (parameter settings are given in Table 2). In our analysis, four solutions are compared:

- *ICP federation*: this is the proposed solution where the game theoretic federation formation algorithm in Section 4 is implemented;
- *No federation*: with this solution the single ICPs are working as stand-alone entities without forming cooperative federations;
- *Device federation*: this solution represents a typical MCC solution where the devices in the system cooperate as independent entities. Specifically, this solution implements the same game theoretic federation solution conceived for the ICPs in Section 4, while considering the single devices as the players for the game instead of the ICPs.

Moreover, we consider a benchmark solution for further comparison, wherein all available ICPs (not only those interested in some tasks) are part of a unique federation and resources available in all their devices are exploited to best match the task requests.

In the first analysis we consider a sample scenario where the number of ICPs in the system is set to 4 and each ICP has 3 devices. Each device is endowed with a minimum of 1 and a maximum of 3 resources, randomly selected among the six types of resources we considered, i.e., [1-10] MFLOPS for computation, [10-100] MBytes of storage capacity, and embedded sensors of four possible types (e.g., humidity, temperature, pressure, movement). The number of task requests generated are reported in Table 2. In particular, we generated task requests so that the total amount of resources available and the total amount of resources requested by the tasks are comparable. This setting guarantees that, in the

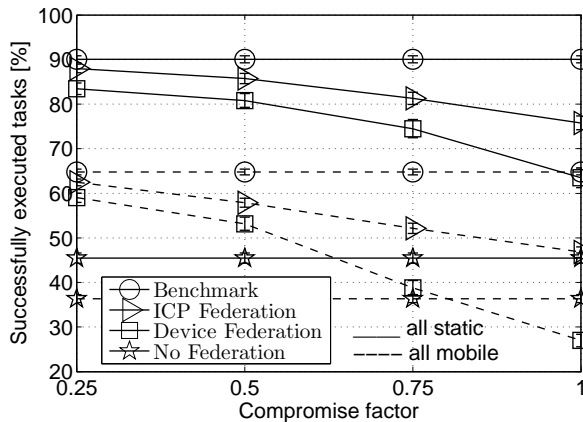


Figure 3: Impact of mobility and compromise factor on the number of successfully executed tasks.

best case, all task requests can be successfully executed. Finally, we considered different values of the compromise factor α .

The first results in Fig. 3 show the percentage of executed tasks when considering all ICPs either being static or mobile and for different values of the compromise factor (other values of % of mobile ICPs bring to results falling in-between the two extreme cases reported in the plots). Results are shown with a 95% confidence interval. As expected, from the plots we note that, for all solutions, mobility adversely affects the number of successfully executed tasks. Another expected result is that the *No federation* and the *Benchmark* solution always show the lowest and the highest performance respectively both for all static ICPs and for all mobile ICPs. An exception exist in case of compromise factor set to one and all mobile ICPs, where the *Device federation* is the worst performing solution. Interesting to notice, always the proposed *ICP federation* solution outperforms the *Device federation*. Moreover, we observe that the higher the value of α , the greater is the importance the players attribute to the cost term in their payoff computation. As a consequence, a lower percentage of successfully executed tasks is obtained when increasing α for the federation based solutions. This testifies to the fact that the single game players are more reluctant to cooperate and to sacrifice their resources to solve tasks in the federation.

The next plots we report in Fig. 4 show the percentage of ICPs being satisfied. The considered scenario is the same as in Fig. 4 (note that the *No federation* is not reported as no comparison can be made to evaluate the ICP satisfaction). The importance of these plots is to show that although the *Benchmark* solution allows to satisfy more tasks, the self-interested players are not satisfied by such a service configuration because they obtain less utility than in the case they work alone. Noteworthy, we wish to recall that our proposed solution guarantees the *stability* of the formed coalition structure. This is also

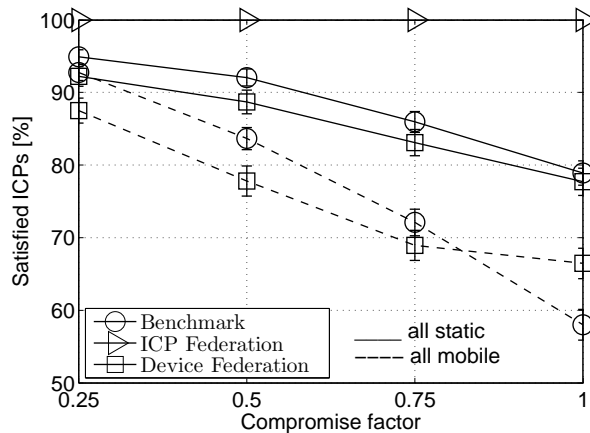


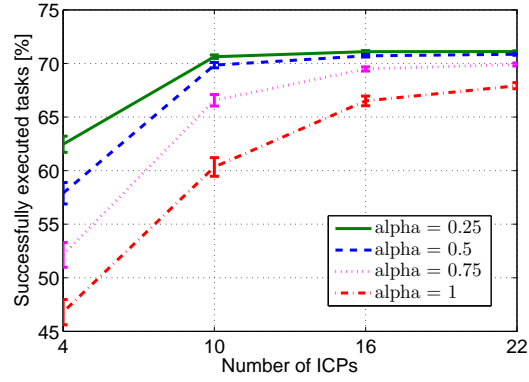
Figure 4: Impact of mobility and compromise factor on the ICP satisfaction.

supported by the 100% of satisfied ICPs in all of the tested cases for the *ICP federation* solution. In particular, the *Benchmark* and the *Device federation* solutions fall below the 57% value and 65% value, respectively, of satisfied ICPs, as the case when all ICPs are mobile and the compromise factor is set to one.

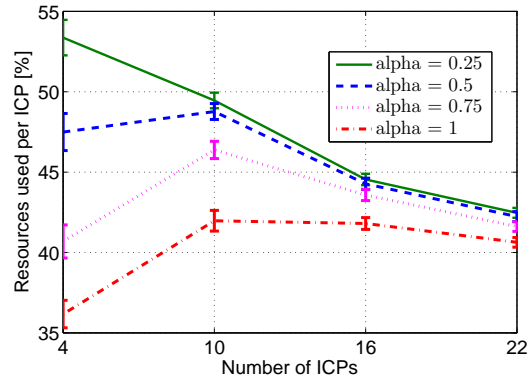
5.1. Analysis for a varying number of ICPs

In this second analysis we investigate on the behavior of the proposed *ICP federation* solution when varying the number of ICPs in the network (the range of ICPs is between 4 and 22). In particular, we consider the results for the case when all ICPs are mobile (for a lower number of mobile ICPs better results are obtained). In Fig. 5 we report the percentage of successfully executed tasks, the amount of resources used per ICP in the federations and the average individual utility per ICP. As it can be observed, for an increasing number of the ICPs in the system one observes no performance deterioration. On the contrary, the proposed solution allows to form cooperative federations of ICPs that increase the performance. In particular, by jointly observing these plots we can derive that higher numbers of ICPs in the network has the following effects: (i) the total amount of executed tasks in the network increases, converging to a value that is dictated by the combination of resources/task requests in the system; (ii) the average individual utility per single ICP in the system increases; and (iii) from 10 ICPs and beyond in the system, the average amount of resources used per ICP decreases. Moreover, we can observe that in all cases for lower values of the compromise factor, all of the performance indicators show better results.

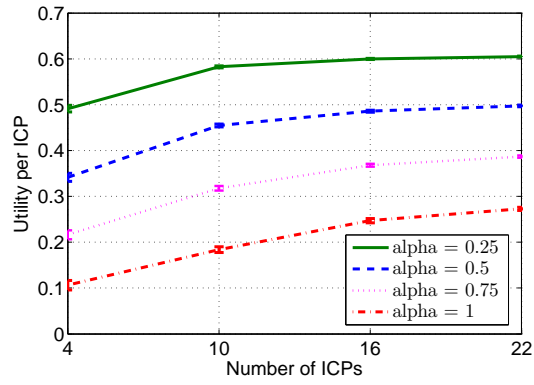
The motivation for this good behavior of the solution may be better understood by discussing the final coalition structure of ICPs in the network. To this aim in Fig. 6 we report information about the number of federations (Fig. 6(a)), the average federation size (Fig. 6(b)) and the size of largest federation



(a) Successfully executed tasks

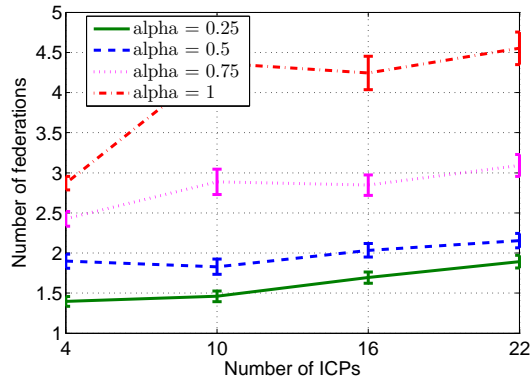


(b) Resources used per ICP

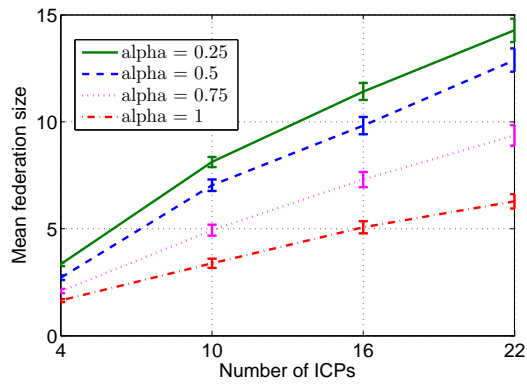


(c) Utility per ICP

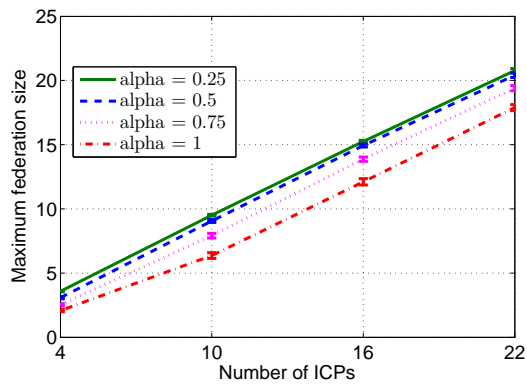
Figure 5: Impact of number of ICPs and compromise factor (all mobile ICPs).



(a) Number of federations



(b) Mean federation size



(c) Maximum federation size

Figure 6: Coalition structure information for varying ICP number and compromise factor (all mobile ICPs).

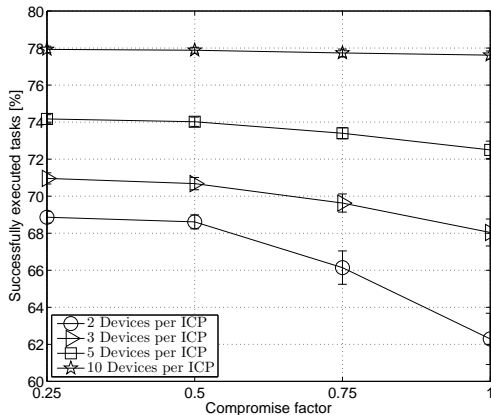


Figure 7: Impact of number of devices per ICPs (22 mobile ICPs).

(Fig. 6(c)) the proposed solution converges to. As the plots show, for a higher number of ICPs in the system the number of federations, the average size of the federations, and the size of the largest federation in the final partition increase. This is the main reason for the improved performance shown in Fig. 5, as in larger coalitions the positive effects of the cooperative behavior have higher impact. Moreover, for lower values of the compromise factor we observe that the maximum federation size in a final partition increases and consequently the number of federations decreases. The reason for this is that the ICPs are less restrictive in joining a federation with lower values of the compromise factor. Finally, it is worth underlining that all final partitions are *stable* and the ICPs are satisfied in being part of their respective federation.

Finally, we have extended the simulative analysis also to a higher number of devices in the scenario and, by this, validate the *scalability* of the proposed solution. More specifically, up to 10 devices per ICP have been considered for a maximum of 22 mobile ICPs, leading to IoT scenarios with up to 220 devices. This number of devices is, in our opinion, a realistic value for the small-scale IoT environments studied in this paper where the edge node is represented, for instance, by cloud-enhanced access points, such as a Wi-Fi AP or a femtocell. The results we have obtained with this analysis are shown in 7 and testify that our proposed solution is scalable, whereas the percentage of successful satisfied tasks increases with the number of devices per ICP. This latter is an expected result, since the more resources are available in a given scenario, the larger are the opportunities to find a required device able to serve a task request in a federation.

6. Future Works and Open Challenges

In the future we plan to address an extension of the model to include scenarios where an ICP represents all the devices owned by a single user, not only those ones carried along with her. Under these conditions, devices of the same ICP can be under coverage of multiple edge nodes (example: PC and laptop at home and car, cell phone, body area sensor network carried around by the owner). In such a situation, it is important to take into account the latency introduced by the communication of different edge nodes and the introduction of migration strategies to move the virtual counterpart of the ICP, namely the ICP controller, from one edge node to another.

We will also make further efforts to thoroughly investigate the energy consumption aspect associated to the proposed MIFaaS paradigm. By accurately modelling it for the execution of each task and for the transmission and networking aspects, suitable task offloading strategies can be studied to improve energy efficiency and prolong the lifetime of battery-equipped devices. In this way, the federation process will have the possibility to perform the mapping between tasks and available devices by appropriately considering the battery energy level of each device and the energy consumption for each task and associated networking and communication activities. To this aim, not only should realistic power consumption models be introduced, but also the task execution and communication times to obtain valid results that bring to real benefits. Furthermore, major modifications should be introduced to the game theoretic model and, in particular, in the utility function for the players. With respect to this, the weighted function should be revised to account also for energy consumption and to allow the players defining their preferences in joining/leaving a coalition by giving priority to either the number of resolved tasks or to energy saving.

Several further researches may enhance the proposed MIFaaS paradigm in future IoT scenarios as discussed in the remainder of this Section.

IoT task allocation algorithm: The IoT application requirements may vary significantly in Quality of Service (e.g., delay, bandwidth, reliability). Therefore, efficient task allocation and resource management offered by the ICPs are of high concern to provide the desired user-perceived quality, i.e., Quality of Experience (QoE). In particular, task mapping algorithms should efficiently face changing network and workload conditions and may also require dynamic support from the external clouds.

Mobility pattern prediction: In an extended multiple-edge scenario, devices being part of an ICP could be under the coverage of different edge nodes and follow different mobility patterns. Mobility pattern predictions for the single devices in realistic scenarios are required to build robust and reliable federations of ICPs. Mobility models should also take into account the type of services offered by single devices, since tasks such as sensing are strictly location-dependent and change of device's position could reduce the relevance of the generated data. Several researchers are working on the issue of user mobility pattern prediction in mobile and wireless environments [47],[48] by adopting also probabilistic and

geometrical analysis based on stochastic geometry [49]. Their found solutions may be an inspiration for further investigation on this aspect. Other solutions based on the Radio signal strength (RSS) for coverage probability may also be of interest in the decision making for an efficient task allocation. In particular, when considering networking aspects and the issues related to the network access, the RSS may influence the performances in terms of latency and energy consumption.

Secure and trusted applications: IoT applications can involve the exchange of personal user information, therefore security and trust are of utmost importance. A weak approach in addressing these issues would also have a strong negative impact on end users' opinion about the proposed framework. To this scope, the network infrastructure assistance may play a fundamental role in our proposed paradigm. Although no specific security algorithms have been integrated in the solution yet, the need for security has been taken into consideration in the design of the involved entities and their interactions. In particular, the edge node acting as an orchestrator of the ICP federation may ensure trusted and secure interactions among ICPs. Moreover, each ICP controller manages the set of own devices within the ICP, so that appropriate security procedures can be implemented to guarantee reliable communication within the single ICP. Further design of solutions for secure and encrypted communications, as well as trusted interactions between ICPs, will be investigated in our future work. In this regard, to enhance trustworthiness in the task allocation process, the recent model of the Social Internet of Things (SIoT) [50] with its notion of object sociality has the potential of building the required trustworthiness among devices [51].

IoT Cloud interoperability: each Telco provider may adopt different implementations of the ICP controller, e.g. by making use of different APIs or semantic languages. As a consequence, ICP controllers registered to different edge nodes may not be able to communicate and cooperate. Indeed, as explained in [52], data interoperability is still one of the main concerns regarding IoT solutions, because different architectures make frequently use of distinct semantic languages for data management and dissimilar protocols for data exchange. Therefore, to enhance the discovery and the orchestration of the resources a common *ontology* is required among the ICPs to guarantee seamless processing and interpretation of data. Very recently [53], FIESTA-IoT achieved interoperability between FIWARE and OneM2M solutions thank to a semantic mediator, which represents a major result in this context.

7. Conclusion

In this paper we have presented Mobile-IoT-Federation-as-a-Service (MI-FaaS), a novel paradigm implemented at the network's edge nodes to support dynamic cooperation among private/public local clouds of IoT devices. To collectively solve a set of task requests, IoT Cloud Providers (ICPs) cooperate by forming federations so that the resources of all involved ICPs can be shared,

while considering rationality and self-interest of the ICPs. To manage the federation formation process, a game theoretic model has been proposed, which converges to a *Nash-stable* solution. A proof-of-concept simulation campaign has been performed to demonstrate that the proposed paradigm can guarantee an increased percentage of served tasks when compared to device-oriented and non-cooperative solutions. Finally, a discussion on open challenges is presented for the extension of the proposed MIFaaS model in wide area scenarios where both local and remote devices of an ICP could be involved in the federation process by relying on a multi-edge infrastructure. Our future work will also consider the implementation of a testbed in realistic IoT environments to further validate the proposed paradigm and support it with experimental results. This experimental approach with a running prototype implementation would allow us to verify the proposed modeling and obtain indications about practical benefits of the developed strategy.

- [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] 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*, 2015.
- [3] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, “Clonecloud: elastic execution between mobile device and cloud,” in *Proceedings of the sixth conference on Computer systems*. ACM, 2011, pp. 301–314.
- [4] E. Cuervo, A. Balasubramanian, D.-k. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, “MAUI: making smartphones last longer with code offload,” in *Proceedings of the 8th international conference on Mobile systems, applications, and services*. ACM, 2010, pp. 49–62.
- [5] J. Soldatos, N. Kefalakis, M. Hauswirth, M. Serrano, J.-P. Calbimonte, M. Riahi, K. Aberer, P. P. Jayaraman, A. Zaslavsky, I. P. Žarko *et al.*, “OpenIoT: Open Source Internet-of-Things in the Cloud,” in *Interoperability and Open-Source Solutions for the Internet of Things*. Springer, 2015, pp. 13–25.
- [6] V. Foteinos and et al., *A cognitive management framework for empowering the internet of things*. Springer, 2013.
- [7] A. Ahmed and E. Ahmed, “A survey on mobile edge computing,” in *10th IEEE International Conference on Intelligent Systems and Control (ISCO)*. IEEE, 2016.
- [8] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, “The case for vm-based cloudlets in mobile computing,” *Pervasive Computing, IEEE*, vol. 8, no. 4, pp. 14–23, 2009.

- [9] U. Shaukat, E. Ahmed, Z. Anwar, and F. Xia, “Cloudlet architectures, applications, and open challenges to deployment in local area wireless networks,” in *Journal of Network and Computer Applications*. Elsevier, 2015.
- [10] Y. Jararweh, A. Doulat, O. AlQudah, E. Ahmed, M. Al-Ayyoub, and E. Benkhelifa, “The future of mobile cloud computing: Integrating cloudlets and mobile edge computing,” *23rd International Conference on Telecommunications (ICT), Greece*, 2016.
- [11] 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.
- [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 13-th ACM international symposium on Mobile Ad Hoc Networking and Computing*. ACM, 2012, pp. 145–154.
- [13] T. Penner, A. Johnson, B. Van Slyke, M. Guirguis, and Q. Gu, “Transient clouds: Assignment and collaborative execution of tasks on mobile devices,” in *Global Communications Conference (GLOBECOM), 2014 IEEE*. IEEE, 2014, pp. 2801–2806.
- [14] N. Fernando, S. W. Loke, and W. Rahayu, “Mobile cloud computing: A survey,” *Future Generation Computer Systems*, vol. 29, no. 1, 2013.
- [15] E. Ahmed, A. Gani, M. K. Khan, R. Buyya, and S. U. Khan, “Seamless application execution in mobile cloud computing: Motivation, taxonomy, and open challenges,” *Journal of Network and Computer Applications*, vol. 52, pp. 154–172, 2015.
- [16] E. Ahmed, A. Gani, M. Sookhak, S. H. Ab Hamid, and F. Xia, “Application optimization in mobile cloud computing: Motivation, taxonomies, and open challenges,” *Journal of Network and Computer Applications*, vol. 52, pp. 52–68, 2015.
- [17] T. Liu, F. Chen, Y. Ma, and Y. Xie, “An energy-efficient task scheduling for mobile devices based on cloud assistant,” *Future Generation Computer Systems*, vol. 61, pp. 1–12, 2016.
- [18] S. Abolfazli, Z. Sanaei, E. Ahmed, A. Gani, and R. Buyya, “Cloud-based augmentation for mobile devices: motivation, taxonomies, and open challenges,” *Communications Surveys & Tutorials, IEEE*, vol. 16, no. 1, pp. 337–368, 2014.
- [19] E. Miluzzo, R. Cáceres, and Y.-F. Chen, “Vision: mClouds-computing on clouds of mobile devices,” in *Proceedings of the third ACM workshop on Mobile cloud computing and services*. ACM, 2012, pp. 9–14.

- [20] T. H. Luan, L. Gao, Z. Li, Y. Xiang, and L. Sun, “Fog Computing: Focusing on Mobile Users at the Edge,” *arXiv:1502.01815*, 2015.
- [21] S. W. Loke, “Supporting ubiquitous sensor-cloudlets and context-cloudlets: Programming compositions of context-aware systems for mobile users,” *Future Generation Computer Systems*, vol. 28, no. 4, pp. 619–632, 2012.
- [22] M. H. Rehmani and A.-S. K. Pathan, *Emerging Communication Technologies based on Wireless Sensor Networks: Current Research and Future Applications*. CRC Press, Taylor and Francis Group, 2015.
- [23] 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.
- [24] L. Barreto, A. Celesti, M. Villari, M. Fazio, and A. Puliafito, “Security and IoT Cloud Federation: Design of Authentication Schemes,” in *EAI International Conference on Cloud, Networking for IoT systems, Lecture Notes of ICST (LNICST)*, 2015.
- [25] K. Habak, M. Ammar, K. A. Harras, and E. Zegura, “FemtoClouds: Leveraging Mobile Devices to Provide Cloud Service at the Edge,” in *IEEE Cloud*, 2015.
- [26] I. Farris, L. Militano, M. Nitti, L. Atzori, and A. Iera, “Federated Edge-assisted Mobile Clouds for Service Provisioning in Heterogeneous IoT Environments,” in *IEEE World Forum on Internet of Things*, 2015.
- [27] R. Kaewpuang, D. Niyato, P. Wang, and E. Hossain, “A framework for cooperative resource management in mobile cloud computing,” *Selected Areas in Communications, IEEE Journal on*, vol. 31, no. 12, pp. 2685–2700, December 2013.
- [28] Y. Ge, Y. Zhang, Q. Qiu, and Y.-H. Lu, “A game theoretic resource allocation for overall energy minimization in mobile cloud computing system,” in *Proceedings of the 2012 ACM/IEEE international symposium on Low power electronics and design*. ACM, 2012, pp. 279–284.
- [29] P. Antoniadis, S. Fdida, T. Friedman, and V. Misra, “Federation of virtualized infrastructures: sharing the value of diversity,” in *Proceedings of the 6th International Conference*. ACM, 2010, p. 12.
- [30] M. Guazzone, C. Anglano, and M. Sereno, “A game-theoretic approach to coalition formation in green cloud federations,” in *Cluster, Cloud and Grid Computing (CCGrid), 2014 14th IEEE/ACM International Symposium on*. IEEE, 2014, pp. 618–625.

- [31] L. Mashayekhy, M. Nejad, and D. Grosu, “Cloud federations in the sky: Formation game and mechanism,” *Cloud Computing, IEEE Transactions on*, vol. PP, no. 99, pp. 1–1, 2014.
- [32] P.-Y. R. Ma, E. Y. S. Lee, and M. Tsuchiya, “A task allocation model for distributed computing systems,” *IEEE Transactions on Computers*, vol. 31, no. 1, pp. 41–47, 1982.
- [33] B. Billet and V. Issarny, “From task graphs to concrete actions: a new task mapping algorithm for the future internet of things,” in *11th IEEE International Conference on Mobile Ad hoc and Sensor Systems*, 2014.
- [34] G. Colistra, V. Pilloni, and L. Atzori, “The problem of task allocation in the Internet of Things and the consensus-based approach,” *Computer Networks*, vol. 73, pp. 98–111, 2014.
- [35] R. Petrolo, V. Loscrì, and N. Mitton, “Towards a smart city based on cloud of things, a survey on the smart city vision and paradigms,” *Transactions on Emerging Telecommunications Technologies*, 2015.
- [36] L. Wang, L. Liu, X. Cao, X. Tian, and Y. Cheng, “Sociality-aware resource allocation for device-to-device communications in cellular networks,” *IET Communications*, vol. 9, no. 3, pp. 342–349, 2015.
- [37] Y. Li, D. Jin, Z. Wang, L. Zeng, and S. Chen, “Exponential and power law distribution of contact duration in urban vehicular ad hoc networks,” *Signal Processing Letters, IEEE*, vol. 20, no. 1, pp. 110–113, 2013.
- [38] E. C.-H. Ngai and P. Gunningberg, “Quality-of-information-aware data collection for mobile sensor networks,” *Pervasive and Mobile Computing*, vol. 11, pp. 203–215, 2014.
- [39] W. Saad, Z. Han, M. Debbah, A. Hjørungnes, and T. Basar, “Coalition game theory for communication networks: A tutorial,” in *IEEE Signal Processing Magazine, Special issue on Game Theory in Signal Processing and Communications*, sep 2009.
- [40] D. E. Knuth, *The Art of Computer Programming, Volume 4, Combinatorial Algorithms, Part 1*. Addison-Wesley Professional, 2011.
- [41] T. Sandholm, K. Larson, M. Andersson, O. Shehory, and F. Tohmé, “Coalition structure generation with worst case guarantees,” *Artificial Intelligence*, vol. 111, no. 1, pp. 209–238, 1999.
- [42] A. Bogomolnaia and M. O. Jackson, “The stability of hedonic coalition structures,” *Games and Economic Behavior*, vol. 38, pp. 201–230, 2002.
- [43] Z. Zhang, L. Song, Z. Han, and W. Saad, “Coalitional games with overlapping coalitions for interference management in small cell networks,” *Wireless Communications, IEEE Transactions on*, vol. 13, no. 5, pp. 2659–2669, 2014.

- [44] M. Guazzone, C. Anglano, R. Aringhieri, and M. Sereno, “Distributed coalition formation in energy-aware cloud federations: A game-theoretic approach (extended version),” *arXiv preprint arXiv:1309.2444*, 2013.
- [45] Y. Li, D. Jin, J. Yuan, and Z. Han, “Coalitional games for resource allocation in the device-to-device uplink underlying cellular networks,” *Wireless Communications, IEEE Transactions on*, vol. 13, no. 7, pp. 3965–3977, July 2014.
- [46] K. R. Apt and A. Witzel, “A generic approach to coalition formation,” *International Game Theory Review*, vol. 11, no. 03, pp. 347–367, 2009.
- [47] W.-S. Soh and H. S. Kim, “Qos provisioning in cellular networks based on mobility prediction techniques,” *Communications Magazine, IEEE*, vol. 41, no. 1, pp. 86–92, 2003.
- [48] W. Wang and I. F. A. Yildiz, “On the estimation of user mobility pattern for location tracking in wireless networks,” in *Global Telecommunications Conference, 2002. GLOBECOM’02. IEEE*, vol. 1. IEEE, 2002, pp. 610–614.
- [49] D. Moltchanov, “Distance distributions in random networks,” *Ad Hoc Networks*, vol. 10, no. 6, pp. 1146–1166, 2012.
- [50] 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.
- [51] 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.
- [52] 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. PP, no. 99, pp. 1–1, 2015.
- [53] FIESTA-IoT, “Federated interoperable semantic iot testbeds and applications,” 2016. [Online]. Available: <http://fiesta-iot.eu/fiesta-achieves-semantic-interoperability-between-fiware-and-onem2m/>