

A Multidimensional Control Architecture for Combined Fog-to-Cloud Systems

Xavi Masip-Bruin¹, Vitor Barbosa Souza², Eva Marín-Tordera¹, Guang-Jie Ren³
Admela Jukan⁴, Jordi Garcia¹,

¹ CRAAX Lab, Universitat Politècnica de Catalunya, 08800 Vilanova i la Geltrú, Spain

² Universidade Federal de Viçosa, 36570 Minas Gerais, Brazil

³ IBM Almaden Research Center, 94002 San Jose, USA

⁴ Technische Universität Braunschweig, 38106 Braunschweig, Germany

{xmasip, eva, jordig}@ac.upc.edu, vitorbs@dpi.ufv.br
gren@us.ibm.com, a.jukan@tu-bs.de

Abstract. The fog/edge computing concept has set the foundations for the deployment of new services leveraging resources deployed at the edge paving the way for an innovative collaborative model, where end-users may collaborate with service providers by sharing idle resources at the edge of the network. Combined Fog-to-cloud (F2C) systems have been recently proposed as a control strategy for managing fog and cloud resources in a coordinated way, aimed at optimally allocating resources within the fog-to-cloud resources stack for an optimal service execution. In this work, we discuss the unfeasibility of the deployment of a single control topology able to optimally manage a plethora of edge devices in future networks, respecting established SLAs according to distinct service requirements and end-user profiles. Instead, a multidimensional architecture, where distinct control plane instances coexist, is then introduced. By means of distinct scenarios, we describe the benefits of the proposed architecture including how users may collaborate with the deployment of novel services by selectively sharing resources according to their profile, as well as how distinct service providers may benefit from shared resources reducing deployment costs. The novel architecture proposed in this paper opens several opportunities for research, which are presented and discussed at the final section.

Keywords: fog computing, combined F2C systems, virtual control architecture.

1 Introduction and Motivation

The unstoppable growth of devices at the edge of the network –including wearables, smartphones, vehicles, sensors, actuators or appliances–, along with the development of distinct network technologies –enabling wireless sensor networks, machine-to-machine (M2M) communication or pervasive computing, just to name a few–, paved the way for the so-called Internet of Things (IoT) [1]. Simultaneously, core

technologies have substantially evolved including enhanced distributed and high performance computing, data center networks or self-manageable resources, among others. Leveraging such technological evolution along with wide network ubiquity, higher network availability and, finally, the severe needs imposed by innovative (foreseen but also unforeseen) services, cloud computing was developed to enable remote requests for service execution, anywhere and anytime with seamless integration of distinct end-user devices [2]. When put together, Cloud and IoT pave the way for deploying new highly demanding services, benefitting from both real-time data collection from devices at the edge and processing power and long-term storage both brought up by cloud providers at the core of the network. Novel scenarios, such as smart cities, smart home, smart transportation or smart agriculture, are remarkably evolving by adopting the smart capacities brought by such a technological deployment. However, the increasing demand for real-time IoT services, such as dependable services in e-Health, traffic control in smart transportation and optimized tracking in Industry 4.0, whose requirements include not just real-time data collection, but also real-time data processing, has put in check cloud computing as the appropriate solution for provisioning real-time sensitive IoT applications. Indeed, the large distance between cloud data centers and end-user devices undoubtedly introduces a considerable latency for remote service execution.

In order to cope with the delay added by the employment of cloud premises, fog computing [3] has been recently proposed aiming at diminishing the network delay, by bringing computing resources close the end-user, through highly virtualized micro data centers (see also [4] for a recent survey on the topic). In fact, with the unstoppable growth of edge devices' capabilities in distinct aspects –processing power, storage, autonomy and connectivity–, the set of services that may be offered by idle edge resources increases significantly, rather than being only considered for data provisioning. Nevertheless, it is worth noticing that the set of services benefitting from fog computing may be limited by the capacities brought by very constrained edge resources as well as by their dynamicity and volatility. Thus, the execution of services demanding high processing capacity, for instance, should not dismiss reliable cloud resources. Therefore, it seems obvious that the role of fog is not to compete with cloud, but to set a global collaborative scenario where services execution, regardless their demands and requirements, may benefit from both, cloud and fog, allocating those resources best suiting specific service demands, be it either at cloud, fog or a combination of both.

Achieving an optimal (fog and/or cloud) resource allocation –empowering QoS-aware service execution, low network load, green computing, and scalability– requires a novel control mechanism intended to considering the characteristics of each resource type, the set of provided services, as well as the end-user service requirements. Recent efforts are being devoted to design a general architecture dealing with fog and cloud resources control, turning into two main directions. An active work is led by the OpenFog Consortium through the recently published OpenFog Reference Architecture [5]. Another approach, referred to as Fog-to-Cloud (F2C), was proposed in [6], aiming at designing a solution for coordinated management of fog and cloud resources through a layered topology, hierarchically organized, enabling parallel service execution in any layer of the envisioned resources topology. More recent references related to the F2C approach propose solutions for

F2C layers distribution ([7], [8]) according to distinct devices characteristics, such as processing capacity, energy availability, mobility profile, and communication technologies, and also evaluate the benefits brought by its deployment [9]. This paper puts the focus on the roots of the F2C architecture and specifically analyzes to what extent a unique, single management architecture may handle any control demand (whatever it be, and whatever it demands) to come from services, scenarios, devices and users. Thus, the question is, “may a unique control architecture support any potential demand?”. The main rationale for such a question is twofold, devices mobility and system heterogeneity. Fig. 1, intended to illustrate the problem, shows a 3-layered F2C resources topology considering both mobility and proximity to the end-user as the main attributes for the F2C layers definition. Two areas are graphically included, preliminary splitting resources according to the geographical distribution, although additional policies must be defined (still an ongoing work). Indeed, the lower layer (fog layer 1 or simply Fog-1) is composed mainly of resources on the move, such as vehicles and smartphones, providing low network latency at the cost of a higher disruption probability. The upper fog layer (fog layer 2 or Fog-2) is composed of permanently or temporarily static resources in a smart scenario whose network latency may be higher than the one perceived in the Fog-1 resources, albeit it is still considerably lower than the cloud communication delay. This layer may embrace both fixed resources in a building and resources provided by cars in a parking lot, for instance. Finally, the upper layer (cloud layer) is composed of reliable resources provided by cloud data centers.

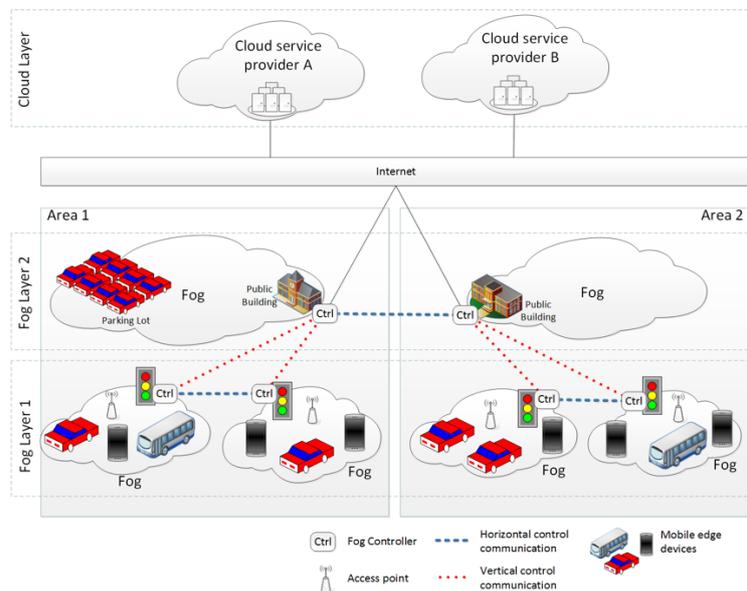


Fig. 1. F2C layered architecture.

Albeit the general F2C topology may follow the characteristics presented in Fig. 1, there is no way to guarantee that such management topology is the most appropriate

to meet the entire set of requirements for any potential service. In fact, looking at the expected high mobility, large heterogeneity, collaborative models based on sharing policies and also assuming that the current world of innovative smart services is only in its infancy, the management topology for a proper global service execution might rely on distinct management configurations to best meet specific service requirements and the existing resources capacities. It is our believe that one logically centralized management and control topology, handling any service to be executed and any real-time resource topology, may not be able to guarantee near-optimal resources selection and services orchestration. Therefore, we envision distinct control topologies able to optimize the management of heterogeneous and dynamic resources, tailored to manage service demands optimally, thus setting a sort of a multidimensional control architecture, set by different views of the real control devices into virtual control instances. This paper is intended to introduce this multidimensional approach for the F2C architecture, mainly emphasizing on:

- Positioning the need for a multidimensional architecture, leveraging virtual control instances and discussing its concept and main benefits;
- Providing the main characteristics of the multidimensional approach regarding control and data plane;
- Presenting research opportunities as well as open challenges for the successful deployment of the proposed multidimensional architecture.

The rest of this paper is organized as follows. Section 2 presents previous works in the literature regarding the decoupling of control and data planes in scenarios with high dynamicity. In Section 3, the multidimensional architecture being proposed in this paper is presented, and some preliminary results are drafted in Section 4. The main research opportunities in this area are presented in Section 5, whilst Section 6 concludes the paper

2 The Roots Endorsing the Multidimensional Approach

This section aims at presenting the distinct inputs that have contributed as a background for the positioning of this multidimensional approach for F2C systems –it could be applied to any strategy designed to managing the full stack of resources from the edge up to the cloud though. One of the main concepts behind the multidimensional architecture is related to the decoupling of control plane from the data (or forwarding) plane, as proposed by Software Defined Networking (SDN). Albeit the big enhancement introduced by the advent of SDN on existing network topologies has been initially limited to scenarios presenting high stability, such as data center networks, several works have foreseen the benefits of SDN adoption in geographically distributed environments presenting wireless communications and high dynamicity, such as VANET or IoT [10]. Authors in [11] propose an SDN architecture for VANETs where control communication may follow distinct approaches including centralized, distributed and hybrid. Whilst centralized and distributed concepts are similar to conventional SDN and VANETs respectively, in the hybrid operation mode, a centralized SDN controller delegates control functions to local agents, such as policy rules and routing protocols' parameters dissemination. The main aim of the proposed architecture is to provide resilience in SDN-based

routing for VANETs. Fog and SDN are combined in [12] in order to deploy a logically centralized SDN controller for orchestrating IoT services in distinct fogs. To that end, an updated view of the underlying resources is required by the SDN controller enabling real-time detection of policies violation, resource reservation, and flow rules dissemination, among others. Therefore, local SDN agents are deployed at each fog node in order to control intra-fog communication through policies obtained by the SDN controller.

The high dynamicity and heterogeneity of the existing resources in terms of computing performance, storage capacity, energy consumption, mobility, reliability and volatility, among others, may demand the specialization of local controllers in order to support the diverse and heterogeneous nature of the service demands. For instance, controllers deployed for the management of services to be executed within vehicles on the move –such as infotainment, traffic control, urban resilience, etc.–, may store the city topology map as well as vehicles position and speed in order to perform accurate handover predictions [12]. On the other hand, controllers handling services demanding communication with environment monitoring resources, such as those deployed through WSN, may keep information about the energy profile of edge resources, hence maximizing service lifespan.

Leveraging SDN, Service-Oriented Architectures (SOA) and fog computing concepts, the Control as a Service (CaaS) concept introduced by [8], aims at using idle resources at the edge of the network –e.g., processing, storage or network resources–, to store and keep an updated view of underlying resources, map service requests into the most suitable resources, and enable efficient inter-controller communications, among others. The proposed strategy aims at enabling control decisions to be taken closer to end-users. Therefore, near-zero delay demanded by real-time services can be successfully achieved from the first steps of the service allocation process, when the edge resources selection and reservation are performed by controllers at the edge of the network –rather than achieving reduced delay only on the service execution and the data transmission. The assessment of the F2C architecture control topology presented in that work has shown the tradeoffs between controllers' capacity, number of controllers, and number of control layers. Although preliminary results have shown the feasibility of the proposed paradigm, several issues are still unsolved, mainly dealing with the dynamic selection of controllers, control topology, and resilience strategies, just to name a few, all fostering new research avenues and thus seeking for future work.

Moving the control decisions to the edge has also been assessed by [13], where authors present a framework whose resource management is sent to the edge rather than letting it to the service providers. The management of resources, which are organized as Mini datacenters (MDCs), is assigned to the Edge Computing Infrastructure Provider (ECIP), which establish contracts with service providers enabling an auction-based edge resource sharing. In that work, the relationship between ECIP and MDC is 1-to-N, i.e., one MDC can be operated exclusively by one single ECIP, what does not meet the conditions of the multidimensional control architecture proposed in this paper.

3 Multidimensional Architecture

In this section we introduce the multidimensional control concept, as envisioned for the F2C architecture, designed to represent distinct control plane instances through overlapped planes, bringing in a key difference with the traditional utilization of a single control plane. For the sake of comprehension, we initially go over the multidimensional representation of the data plane. Later in this section, we dig into the multidimensional control architecture through the introduction and analysis of distinct scenarios. It is worth mentioning that even though the proposed multidimensional control solution is designed to be applied on a F2C system, the rationale behind the concept supports its deployment on any control strategy mixing highly demanding services, mobility and resources heterogeneity.

3.1 Data Plane

Nowadays, Service-Oriented Architectures (SOAs) leverage the diversity of available resources –be it either distributed or centralized and offering distinct capabilities– to support effective services execution. To that end, resources may be allocated to either a whole service or a simple task (part of a more complex service), the latter requiring an orchestration strategy to enable the successful execution of the whole service. This task aggregation process may consider distinct execution strategies, either sequential, parallel, or a combination of both. Therefore, assuming the fact that services may be composed of distinct service chains, two or more services may be represented by distinct logical planes, each one with the respective data path topology, as illustrated by *Service A* and *Service B* in Fig. 2. On one hand, *Service A*, which has its data path presented in foreground, may be a general representation of services responsible for obtaining and preprocessing data collected from devices at the edge of the network. On the other hand, *Service B*, which has its data path presented in background, may represent services requiring resources with high processing capability in order to generate a detailed and low-delay analysis of data stored in distinct datasets.

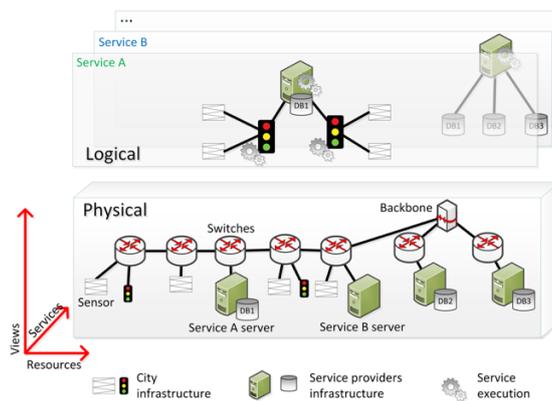


Fig. 2. Multidimensional data plane view for the execution of distinct services.

For instance, let's consider *Service A* collects data through several sensors deployed in a smart city obtaining air pollution data, which is later aggregated by distributed computing resources deployed in city traffic lights, generating the input for the creation of a pollution map for a neighborhood in real-time. The pollution map may be spread to subscribed users or third-party services according to a certain policy (e.g., when reaching a threshold predetermined according to their profiles), so that subscribers may take the expected actions. Service subscribers may include, for instance, applications related to e-Health, Smart Transportation, Smart Environment and Smart Industry, each one requiring specific update intervals and map coverage area. On the other hand, *Service B* aims at selecting the best route for vehicles according to several real-time information and user preferences, including reduction of route time, traffic jams, pollution, tolls, number of traffic lights (reduction of overall number of stops), and number of public services vehicles in the selected route (such as garbage collector trucks), among others.

3.2 Control Plane

Albeit scalability concerns in conventional SDN networks have been largely discussed in the recent literature (see for example [14] and [15]), control mechanisms demanded by novel F2C systems envision the coordinated management of the whole set of resources in order to optimally map available resources into the services requirements. However, the conventional SDN control plane is responsible for the communication among distinct networks, that is, the specific information regarding end-points of each network is usually not considered when setting a path between two networks and, consequently, resources information kept by controllers is limited to conventional network information, such as network addresses, interfaces, and costs, among others. This condition may substantially impact on scenarios requiring edge devices information, such as fog computing and F2C systems. Indeed, when putting together fog computing and SOAs, novel IoT services may be deployed, leveraging not only network aspects (network communication technologies, available network interfaces, bandwidth, etc.), but also the set of resources available at the edge of the network, including for example processing and storage resources. In fact, considering such additional set of resources enables the execution of the desired network services, but also new highly interesting features, such as the offloading of services deployed on end-user devices.

Hence, in IoT scenarios leveraging edge approaches (fog, edge, F2C), additional control information must be also contemplated to both optimally map services into resources and enable new performance functionalities. Thus, novel control architectures must be designed, considering the whole set of information needed to efficiently support services demands while optimizing resources availability. The required additional information must represent characteristics inherent to each resource type, such as:

- Processor: architecture, clock rate, number of shared cores, cache size.
- Storage: type, capacity, read/write velocity.

- Sensor/actuator: type of data/action produced by the device as well as the inherent characteristics of each type, such as range, resolution, sensitivity, just to name a few.

Distinct control topologies have been studied aiming at the deployment of SDN networks, including centralized, decentralized and hierarchical topologies with distinct number of layers [15]. Each topology may present benefits and drawbacks regarding deployment simplicity, scalability, cost, response time, or manageability, just to name a few. We argue that the control scenario envisioned by F2C systems does not fit into a single control topology approach. Moreover, the issue is not only what the control topology will be (centralized, decentralized, hierarchical) but also whether a unique one may cover the whole set of control needs and specifications. Indeed, recognizing the diversity in services demands, the constraints brought by adopting resources mobility and the increasing heterogeneity of resources, we definitely argue that using one single and unique control topology for the management of the diversity of service categories enabled by smart scenarios in an IoT world turns to be unfeasible –the mapping of services requirements into the most suitable resources must be met taking into consideration several resource characteristics, such as the aforementioned ones. Therefore, the envisioned scenario enforces the adoption of new control approaches, such as the creation of specific resources lookup tables, containing resources able to support distinct service requirements.

Aligned to this trend, we push for the creation of distinct control plane instances (kind of control virtualization), each one managing specific resources according to a distinct set of characteristics. For instance, one control instance may manage resources able to provide high performance computing whilst another one may manage resources able to provide green computing. Several approaches must be considered aiming at the optimization of control planes deployment. It is worth mentioning that the relationship between control and resources is not exclusive, i.e., one resource characteristic may match more than one control instance. Each control plane instance may be composed by a distinct set of controllers, thus presenting distinct control topologies intended to optimally manage the underlying resources, thus setting different control views or dimensions (i.e., multidimensional control). In this paper, we follow the work done by the EU mF2C project [16] intended to design a control architecture for F2C systems, and thus we adopt the naming defined within the project. Accordingly, the communication among controllers and underlying resources is done by means of novel elements, referred to as Agents [17], whose deployment on real devices may be performed either as an application download – executed on each resource–, or as an ad-hoc light functionality built on devices with higher simplicity, such as some sensors. The Agent is the element responsible for running all required control functionalities on all the devices participating in the F2C system and will be deployed as different “suites” according to the target device’s hardware (it looks obvious that a deployment on a laptop will not be the same than in a Raspberry Pi controlling a sensor in a city). The Agents are organized into a hierarchy, according to the hierarchical view of the F2C architecture (see Fig.1). Each fog belonging to a Fog Layer will select one Agent (i.e., device) to act as the Leader,

taking over the main control responsibilities for its area of coverage and included elements (see the traffic lights in Fig.1 for Fog Layer 1). The policies designed to handle the Leader selection process are out of the scope of this paper and its design is actually an ongoing work for the authors.

In Fig.1 we show a single control topology, where Leaders are deployed at the traffic lights, and the fogs (two per area) are set meeting a specific policy or a set of deployed rules, yet to be defined. We pose the fact that such a static control topology might not suit all potential demands coming from all services to be executed and available resources. In fact, this is the main rationale to suggest the multidimensional control view, where the control topology may vary to optimally suit resources capacities and services demands. In short, a device may play as Leader for a set of services and as a normal Agent for another one. This means that the Agent software must support such a multidimensional view, defining policies and strategies to guarantee the best control topology for any service.

In order to illustrate the expected behavior for the envisioned multidimensional control approach, we introduce three distinct scenarios, namely different companies, different SLAs and finally the Control as a Service. For the sake of understanding and to perfectly define the different control roles devices may play, we will keep using the terminology and naming used in the mF2C project, as defined previously in this section.

Distinct companies employing shared resource. This scenario poses a smart city putting together different infrastructure components consisting of both the own city infrastructure and the one offered by the envisioned sharing model. First, we consider a smart city whose IoT infrastructure is already deployed and consequently is made available for service providers. Second, beyond the resources deployed by the city, the set of available resources might also include idle resources from users' devices, in a sharing model, where users willing to contribute to the whole set of resources may participate in such a collaborative framework. Therefore, we may assume that the infrastructure to be deployed by a service provider to enable services execution at the available edge resources would simply consist in the resources where the Leader would be deployed at, for a particular set of services –and this would only happen when the service provider wants to have a strict and total control on the services, i.e. a sort of “proprietary” control. It is also worth noting that the usage of shared resources by distinct companies opens opportunities for new business models both between distinct companies and between companies and users. In this scenario, contracts established among parties must define which companies can make use of edge devices, resources provided, priorities, prices, policies, etc. This scenario can be certainly extended for different service providers turning into the distinct control instances depicted in Fig. 3. We may also see in Fig.3, that whilst several physical resources –e.g., sensors–, are shared by distinct providers, some of them are only available for one of the service providers –e.g., traffic lights–, according to contracts previously established by parties. On the other hand, resources initially available for both providers, such as the yellow cab, must implement some sort of resources monitoring function, since for example the complete utilization of its shared resources

by Company A (see Fig. 3) would make it not available for Company B. Hence, distinct control strategies must be sought considering scalability, required table accuracy, business models, overhead, etc.

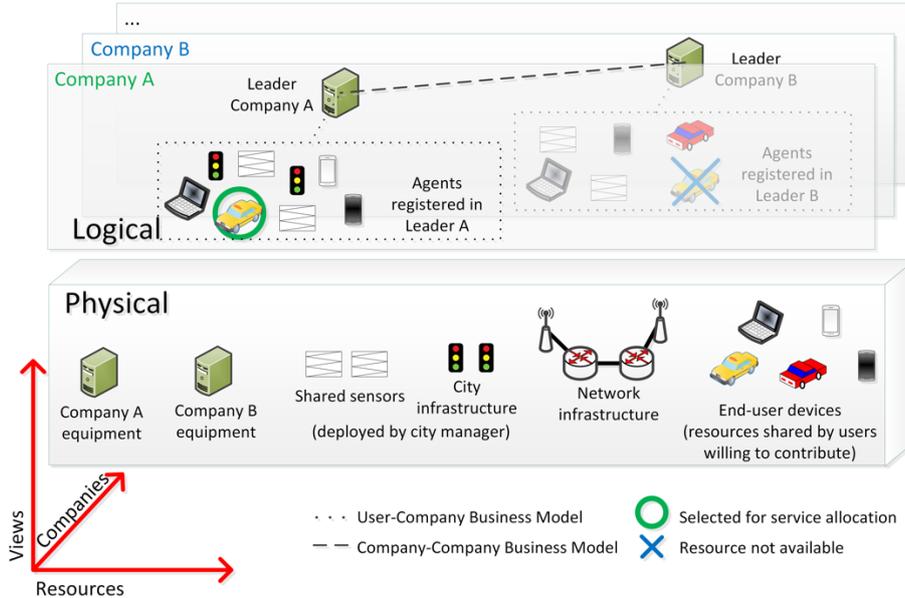


Fig. 3. Multidimensional control plane for management of shared resources by distinct companies.

Distinct SLA Provisioning. Different from the first scenario, this scenario shows the benefits obtained by a single service provider (company) when deploying multiple control instances intended to meet different SLAs in the resources provisioning process. Since the mapping of suitable resources for each service type may require distinct control data –such as resources characteristics specific for each individual service demands–, the deployment of Leaders able to manage a large set of resources producing huge volume of control data in a specific area turns to be non realistic. This means that different Leaders should have to be deployed in a specific area, what undoubtedly would require Leaders communication to keep a synchronized view of the underlying resources, driving a non negligible communications overhead. However, the deployment of distinct control instances managing resources and providing distinct QoE –meeting the required SLAs–, may drastically reduce the number of resources that a Leader needs to manage. For instance, distinct control instances may manage resources demanding distinct requirements, such as green computing, high security communication, high performance computing, and free usage, as illustrated side by side in Fig. 4 for the sake of simplicity. It is also worth noting that distinct topologies for Leaders may be assumed for each SLA, since several factors may be considered in the topology definition, such as amount of underlying resources, their categories, capacities, etc. Moreover, albeit Agents may be

controlled by more than one Leader, resources lookup table in the Leader may be considerably reduced, hence, easing the synchronization among Leaders. Finally, we must also remark that in this second scenario, this synchronization also brings benefits when considering the fact that Leaders are not competing –unlike the first scenario, where competitors may restrict smart agents inter-communication.

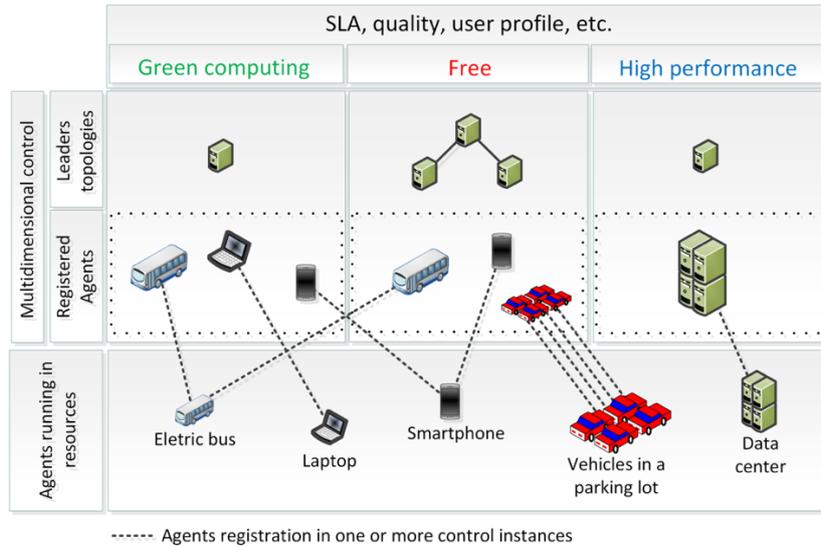


Fig. 4. Multidimensional control plane for provisioning of resources according to SLAs.

Control as a Service. As described in the previous section, the CaaS concept, introduced in [8], aims at using devices at the edge of the network as controllers – leveraging fog computing concepts– in order to bring control decisions closer to the end-user, thus enabling real-time resource selection for sensitive IoT services execution. Unfortunately, the limited capacity inherent to edge devices makes usually unrealistic such devices to simultaneously run distinct services and, more important from a control perspective, to play different control roles, i.e., Agent or Leader for different services. Therefore, the categorization of resources, followed by the creation of distinct resource databases compliant to distinct service requirements, will clearly show the capacity to use constrained resources –in terms of both processing and storage–, to play as Leaders for specific services. This databases categorization will make resource selection and provisioning to be performed with a reduced latency, due to the reduced resources database size, constituted exclusively by resources able to provide the services managed by that Leader.

In such scenario, for each control instance, Agents deployed at edge devices must use strategies for the selection of the most suitable device to play the Leader role according to service characteristics and broadly speaking, according to a policy or strategy to be defined to that end. It must be remarked that, as the control plane may

be formed by multiple control instances, the device playing as the Leader in one instance, may or may not be used as Leader for other control instances. Therefore, an edge device selected as Leader for a service may use part of the shared resources to run the Leader and further use idle shared resources for the execution of other services. This scenario is shown in Fig. 5, where edge devices highlighted with green and blue circles are selected as Leaders for service A and B respectively. Analyzing the logical view of the distinct control topologies, two key aspects deserve special attention. First, each service makes use of resources provided by distinct devices, according to their suitability to execute that service, regardless where the resources are. Second, as previously introduced, devices serving as Leader for a particular service may play a different role or simply share resources for other services, for example, the traffic light serving as Leader for service A can further share idle resources for service B provisioning. Finally, it is worth noting that the Agent (i.e., a functionality embedded in the Agent software) should be responsible for handling the sharing of local resources for service execution as well as their utilization as Leader.

4 Preliminary Results

In order to present a preliminary assessment of the concepts presented in this paper, two distinct experiments are carried out in an in-lab testbed deployed at the lab. The two proposed experiments, referred to as distinct SLA provisioning and CaaS, are inferred from the set of scenarios introduced in section 3.2. The first experiment is based on the scenario where distinct SLA are provided by a company through the deployment of distinct control instances, each leveraging the most suitable resources.

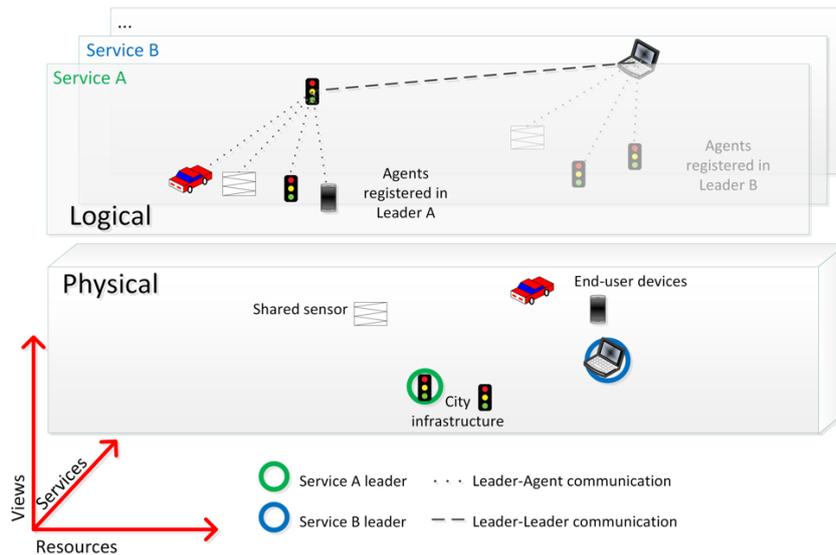


Fig. 5. Control as a Service provided by edge devices employed as leaders for distinct services.

As previously described, the resources are shared among distinct leaders. Thus, an efficient allocation demands an updated view of the underlying resources by all leaders. In this experiment, we assume two static leaders (deployed by one Service Provider) presenting wired connectivity. Both leaders can communicate to synchronize the resources allocation, thus enabling each leader to have an updated view of the shared resources usage. In this approach, we consider that, upon receiving a service request, a leader relies on the current view of underlying resources and selects a set of the most suitable ones for the service provisioning, ordered according to their suitability –which relies on both the service and resources categorization and the previously determined mapping policies. In the next step, the leader establishes wireless communication with the first resource –i.e., the most suitable one– of the ordered set, asking for the required resources reservation and keeps waiting for a reply. In case of failure or negative reply, the procedure is repeated for the next resource of the ordered set, and so on. In case of positive reply, the second leader is informed about the allocation of the shared resource –which may include additional information, such as the estimated allocation time, according to the accepted service categorization.

It is worth noting that, as described by the second scenario included in this paper, each leader in this experiment constitutes a distinct control dimension, where underlying resources may or may not be shared between dimensions. Fig. 6 compares the presented approach and a single dimensional control topology for the first scenario in terms of the processing delay, showing a considerably lower request processing latency for the multidimensional approach proposed in this paper.

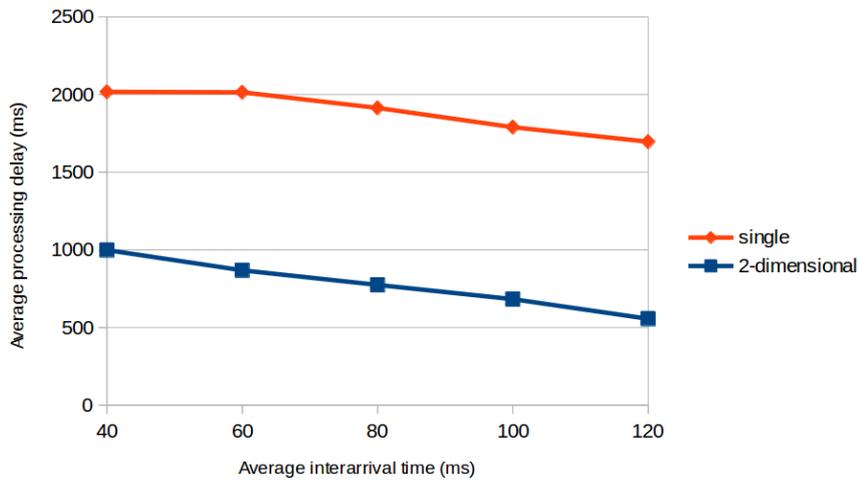


Fig. 6. Comparison between a single and 2-dimensional control for the SLA scenario.

In the second experiment, the CaaS scenario (third use case in section 3.2) is considered. Assuming that the policy to select the Leaders is out of the scope of this paper, two aspects are considered affecting the Leaders selection process. First, in this scenario shared resources at the edge are utilized for the deployment of Leaders.

Second, we assume 2 devices are deployed to play the Leader role with the aim of improving services execution (i.e., two control dimensions). Hence, we assume the first Leader is used for processing the requests from the early beginning of the experiment execution, whilst the second Leader is activated by the first one, according to 2 distinct policies, highlighting the differences brought in when including the multidimensional concept, defined as follows:

- **Forward:** Policy used by the first Leader when it cannot process a received request due to its limited capacity. Hence, the received request is forwarded to the second Leader, which may either accept or deny the request for a service execution according to its current load (additional policies may be added here to enrich this decision). Notice that we consider only 2 Leaders in the proposed experiment, thus impeding further additional request forwarding. This policy does not implement the multidimensional concept, since all Leaders must be ready to run any service and the decision of acceptance is based exclusively on the load of the Leader in terms of amount of concurrent requests. In other words, each Leader should be aware of all the underlying resources offering the services required by the requests received by the first Leader.
- **Split:** This policy considers a multidimensional view where distinct services may leverage distinct control topologies. In this approach, a Leader may decide to accept a particular service or a specific set of services. Therefore, if a Leader cannot handle the management of requests from a new service type, a new Leader is selected among the Agents through an election mechanism (yet to be defined). The new Leader, then, defines a new control topology, taking into account the service requirements and the set of available resources, responsible for the service provisioning. To that end, different approaches may be applied (yet an ongoing work), such as, for example, broadcasting welcome messages containing relevant information about the service and waiting for compliant Agents willing to share resources for that particular service and thus willing to join the new control instance, further enabling service clients to know the leader responsible for the management of that service type.

In addition to these two approaches and for the sake of comparison, a single topology approach is further deployed, in order to analyze the obtained results when no extra Leaders are deployed. In the single approach, no second Leader is considered, hence when the first Leader cannot handle a new request (due the limited capacity of the Leader), the request is denied. Therefore, if a service request does not go through, the service client employs an exponential-back off-based strategy for retrying the service request. It is worth noting that Forward and Split strategies do not guarantee the successful reservation of edge resources, hence, the exponential-back off retransmission scheme may be employed by clients each time a service request is not accomplished, regardless the policy employed by Leaders.

The capacity of each Leader is defined regardless the utilized approach. Therefore, the maximum capacity of a Leader is set as the maximum amount of service requests it is willing to process concurrently. Indeed, we consider, for simplicity, that all service types have the same complexity for resource selection. Therefore, as shown in Fig. 7, for each comparison the capacity of all leaders used is the same. The analysis

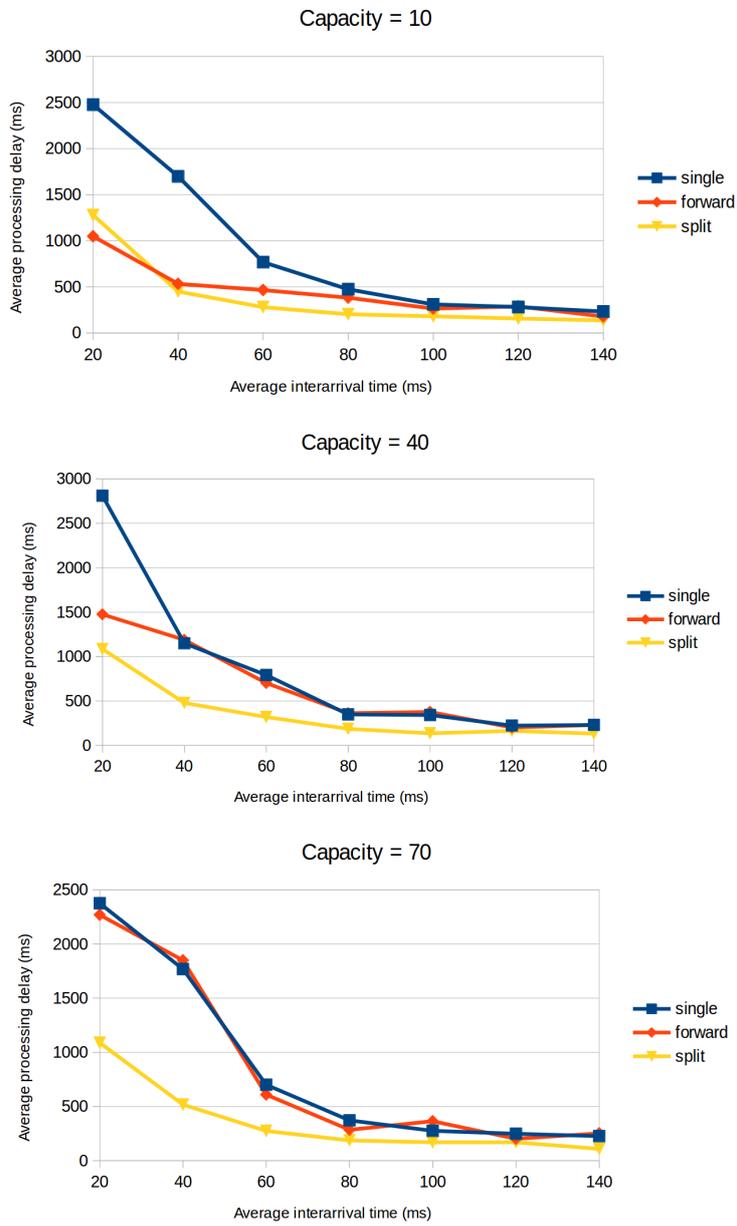


Fig. 7. Comparison of distinct strategies for processing service requests for distinct leader capacities: (a) 10, (b) 40, and (c) 70 concurrent requests.

of the plots in Fig. 7 shows that the best results in terms of delay for resource reservation (average request processing delay) into distinct scenarios set by playing with both the amount of concurrent requests and the requests interarrival time, are obtained when deploying a two dimensional control (split). Furthermore, the increment of capacity of the Leaders (in terms of amount of service requests) does not result in a reduced average request processing delay. This may be justified by the fact that the constrained processing capacity of the Leaders, along with the large number of requests, leads to a high competition and overload of resources used for processing the received requests. Moreover, the increment of the capacity on the Leaders turns into a higher delay when forwarding requests. This is justified by the fact that, with higher capacity, the split strategy reduces the requests forwarding rate, which makes the Leader's behavior to tend to the one presented by the single approach.

5 Opportunities and Challenges

The deployment of the multidimensional control architecture introduced in this paper raises several challenges for its successful deployment. In this section, we assemble the described challenges in order to provide distinct opportunities for future research in this topic, described as following.

In a multidimensional architecture, where shared resources are managed by Agents in distinct dimensions, the scalability assessment is crucial. Indeed, once an Agent acting as Leader selects a resource for service execution, and since the latter can be initially available for more than one Leader, distinct strategies to keep an updated view of resources in distinct Leaders must be assessed.

Since distinct control topologies—such as centralized, distributed and hierarchical—present advantages and disadvantages, strategies for the topology definition for distinct dimensions according to service needs and available resources are required, enabling the optimal deployment of Leaders, minimizing signaling and latency whilst enabling control decisions to be taken closer to end-user making use of updated resources information.

In scenarios where Leaders are dynamically assigned, strategies for runtime selection may be assessed. It is worth mentioning that distinct strategies may be employed for Leader selection considering the layer they are located in control topology, service to be provisioned, amount of Agents to be controlled and their characteristics, among others. In addition, Leaders coordination is an added challenge when considering both intra and inter-control instances coordination.

The deployment of distinct Leaders for the provisioning of distinct services yields new challenges regarding the knowledge of available Leaders by clients. Alternatives must be defined in order to enable each client to discover which Leader is managing resources able to provide the required service. Whilst solutions such as the deployment of brokers may be effective, the added delay must be assessed, especially for highly sensitive services.

The deployment of such a F2C collaborative model fuels the establishment of novel business models not only among service providers, but also between service

provider and clients. For the latter, SLA between each pair client-provider may comprise expected QoS, user preferences regarding shared resources, schedules, privacy, and other preferences that shall be available in user profile. For the former, besides directives for sharing private resources, an SLA may include rules for data sharing while respecting clients' preferences, such as privacy. The definition of novel business models is the basis for the successful deployment of such collaborative model.

As Agents are responsible for managing resources available at edge devices, its implementation shall require the definition of policies to enable proper resource allocation according to preferences defined at user profile.

In traditional host-oriented networking, an SDN controller does not need to have knowledge about the edge devices. Rather, the controller keeps only information regarding forwarding devices topology and, among others, it can define the switches (forwarding devices) that should be used in order to establish communication between two distinct networks. On the other hand, next generation service-oriented IoT applications will require the edge resource selection according to the services offered by them. Moreover, the amount of information regarding the edge resources whose controllers should keep will increase according to the amount of offered services.

Several security concerns must be considered in such a collaborative model. That includes but is not limited to privacy, authentication, access control, identity management, integrity, and availability.

Finally, besides the challenges arisen from the deployment of the proposed multidimensional architecture, several challenges are still not completely solved by considering F2C systems. Therefore, resources discovery, resources monitoring, devices tracking, service allocation, efficient services orchestration, or optimal service-resources mapping, are just some of the research topics that still deserve efforts in order to enable the successful deployment of F2C systems.

6 Conclusion

In the upcoming years, new business models shall arise based on collaborative models where mobile and non-mobile end-users will share idle resources whilst distinct service providers will benefit not only from end-users resources but also from resources deployed on the ground (cities, transport systems, etc.). In this paper, we raise concerns about deploying one single control topology able to provide optimal management of the tremendous amount of envisioned shared resources, while providing distinct QoS requirements according to distinct SLAs for the large set of potential services. Aligned to this concern, this paper positions a multidimensional control architecture for novel combined Fog-to-Cloud (F2C) systems, as a potential solution for a service-tailored management of the devices deployed at the edge of the network. This multidimensional architecture envisions the coexistence of distinct control plane instances enabling optimal management of available resources while fulfilling QoS requirements regarding deployed services and end-user profiles. Through three distinct scenarios, the multidimensional control concept is presented in

a comprehensive manner, and is later evaluated in two of them, intended to highlight its potential benefits. Nevertheless, we reinforce the fact that, due to the novelty of the proposed concept, the multidimensional concept opens many different research avenues, opportunities and challenges –most of them listed in the last section of this paper to provide the reader with a complete picture of the overall scenario–, that must be addressed for a successful deployment of the proposed concept.

Further work will go to many directions, including security provisioning (designing an architecture responsible for providing security to the whole set of F2C systems, considering their specific characteristics, i.e., mobility, heterogeneity, etc.), deployment in specific verticals (including health and vehicular systems) and the design of clustering policies using ML strategies (to identify the proper solution to optimize resources consumption and services performance).

Acknowledgment

This work was supported by the H2020 EU mF2C project, ref. 730929 and for UPC authors, also by the Spanish Ministry of Economy and Competitiveness and the European Regional Development Fund under contract RTI2018-094532-B-I00.

References

1. 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," in *IEEE Wireless Communications*, vol. 23, no. 5, pp. 10-16, October 2016. doi: 10.1109/MWC.2016.7721736
2. J. Gubbi, R. Buyya, S. Marusic and M Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," in *Future Generation Computer Systems*, vol. 29, pp. 1645–1660, September 2013. doi: 10.1016/j.future.2013.01.010
3. F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog Computing and its Role in the Internet of Things," *MCC '12 Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. ACM, 2012. p. 13-16. doi:10.1145/2342509.2342513
4. P.Hu, S.Dhelim, H.Ning, T.Qiu, "Survey on fog computing: architecture, key technologies, applications and open issues", *Journal of Network and Computer Applications*, Vol. 98, pp. 27-42, November 2017
5. OpenFog Consortium, "OpenFog Reference Architecture for Fog Computing", February 2017. https://www.openfogconsortium.org/wp-content/uploads/OpenFog_Reference_Architecture_2_09_17-FINAL.pdf
6. X. Masip-Bruin, E. Marín-Tordera, G. Tashakor, A. Jukan and G. J. Ren, "Foggy clouds and cloudy fogs: a real need for coordinated management of fog-to-cloud computing systems," in *IEEE Wireless Communications*, vol. 23, no. 5, pp. 120-128, October 2016. doi: 10.1109/MWC.2016.7721750
7. X.Masip-Bruin, E.Marín-Tordera, A.Jukan, G.J.Ren, "Managing Resources Continuity from the Edge to the Cloud: Architecture and Performance", *Future Generation Computer Systems*, Vol. 79, Part 3, February 2018

8. V. B. Souza, A. Gómez, X. Masip-Bruin, E. Marín-Tordera, and J. Garcia, "Towards a Fog-to-Cloud Control Topology for QoS-Aware End-To-End Communication," 2017 IEEE 25th International Symposium of Quality of Service (IWQoS), Vilanova i la Geltrú, Barcelona, 2017.
9. W. Ramirez, X. Masip-Bruin, E. Marín-Tordera, V. B. C. Souza, A. Jukan, G. J. Ren, O. González de Dios, "Evaluating the Benefits of Combined and Continuous Fog-to-Cloud Architectures", *Computer Communications*, Vol.113, pp. 43-52, November 2017
10. K. Sood, S. Yu and Y. Xiang, "Software-Defined Wireless Networking Opportunities and Challenges for Internet-of-Things: A Review," in *IEEE Internet of Things Journal*, vol. 3, no. 4, pp. 453-463, Aug. 2016. doi: 10.1109/JIOT.2015.2480421
11. I. Ku, Y. Lu, M. Gerla, R. L. Gomes, F. Ongaro and E. Cerqueira, "Towards software-defined VANET: Architecture and services," 2014 13th Annual Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET), Piran, 2014, pp. 103-110. doi: 10.1109/MedHocNet.2014.6849111
12. S. Tomovic, K. Yoshigoe, I. Maljevic, and I. Radusinovic, "Software-Defined Fog Network Architecture for IoT," in *Wireless Personal Communications*, vol. 92, no. 1, pp. 181-196, 2017. doi: 10.1007/s11277-016-3845-0
13. J. Xu, B. Palanisamy, H. Ludwig and Q. Wang, "Zenith: Utility-aware Resource Allocation for Edge Computing," in *IEEE Edge 2017*, 25-30 Jun 2017, Honolulu, Hawaii
14. S. H. Yeganeh, A. Tootoonchian and Y. Ganjali, "On scalability of software-defined networking," in *IEEE Communications Magazine*, vol. 51, no. 2, pp. 136-141, February 2013. doi: 10.1109/MCOM.2013.6461198
15. J. Hu, C. Lin, X. Li and J. Huang, "Scalability of control planes for Software defined networks: Modeling and evaluation," 2014 IEEE 22nd International Symposium of Quality of Service (IWQoS), Hong Kong, 2014, pp. 147-152. doi: 10.1109/IWQoS.2014.6914314
16. mF2C project at <http://www.mf2c-project.eu>, accessed on-line May 2019
17. Deliverable 2.6, mF2C EU project, at <http://www.mf2c-project.eu/wp-content/uploads/2017/06/mF2C-D2.6-mF2C-Architecture-IT-1.pdf>, accessed online May 2019