



Article

Cognitive Load Balancing Approach for 6G MEC Serving IoT Mashups

Barbara Attanasio ^{1,*}, Andriy Mazayev ^{2,†}, Shani du Plessis ^{2,†} and Noélia Correia ^{2,†}¹ Dipartimento di Ingegneria Elettrica, Elettronica ed Informatica (DIEEI), Università di Catania, 95125 Catania, Italy² Centre for Electronics, Optoelectronics and Telecommunications (CEOT), Universidade do Algarve, Campus de Gambelas, 8005-139 Faro, Portugal; amazayev@ualg.pt (A.M.); sdpleissis@ualg.pt (S.d.P.); ncorreia@ualg.pt (N.C.)

* Correspondence: barbara.attanasio@phd.unict.it

† These authors contributed equally to this work.

Abstract: The sixth generation (6G) of communication networks represents more of a revolution than an evolution of the previous generations, providing new directions and innovative approaches to face the network challenges of the future. A crucial aspect is to make the best use of available resources for the support of an entirely new generation of services. From this viewpoint, the Web of Things (WoT), which enables Things to become Web Things to chain, use and re-use in IoT mashups, allows interoperability among IoT platforms. At the same time, Multi-access Edge Computing (MEC) brings computing and data storage to the edge of the network, which creates the so-called distributed and collective edge intelligence. Such intelligence is created in order to deal with the huge amount of data to be collected, analyzed and processed, from real word contexts, such as smart cities, which are evolving into dynamic and networked systems of people and things. To better exploit this architecture, it is crucial to break monolithic applications into modular microservices, which can be executed independently. Here, we propose an approach based on complex network theory and two weighted and interdependent multiplex networks to address the Microservices-compliant Load Balancing (McLB) problem in MEC infrastructure. Our findings show that the multiplex network representation represents an extra dimension of analysis, allowing to capture the complexity in WoT mashup organization and its impact on the organizational aspect of MEC servers. The impact of this extracted knowledge on the cognitive organization of MEC is quantified, through the use of heuristics that are engineered to guarantee load balancing and, consequently, QoS.

Keywords: 6G; complex network theory; microservices; multi access edge computing; multiplex networks; QoS; task offloading; web of things



Citation: Attanasio, B.; Mazayev, A.; du Plessis, S.; Correia, N. Cognitive Load Balancing Approach for 6G MEC Serving IoT Mashups. *Mathematics* **2022**, *10*, 101. <https://doi.org/10.3390/math10010101>

Academic Editor: Daniel-Ioan Curiac

Received: 1 December 2021

Accepted: 24 December 2021

Published: 28 December 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The forthcoming 6G will attempt to rewrite the communication networks' perspective, focusing on a paradigm shift in the way technologies and services are conceived, integrated and used. The 6G technology has the ambition to provide new directions for dealing with future network challenges. It will address the constraints and the performance requirements of emerging applications and services, many of which have increasing resource needs, through innovative approaches [1,2], such as complex network and bio-inspired methods [3–6]. A context in which this will be quite prominent is the application of the Internet of Things (IoT) to smart cities, which is now attracting attention from both academia and industry. Such environments frequently become difficult to manage because networked Things and people are highly dynamic. One step that can be taken to make this management easier, and to effectively address certain needs, is to adopt efficient resource discovery and access mechanisms. The Web of Things (WoT) has emerged from such a need. WoT effectively allows Things to become Web Things that are accessible via Representational State Transfer (REST) Application Programming Interfaces (APIs) [7].

More precisely, the idea of the WoT is to reuse and leverage widely popular Web protocols, standards and blueprints, to make data and services offered by objects accessible to Web developers. Things expose their data and services as Web resources, allowing reading (e.g., temperature) and/or updates (e.g., trigger an actuation) by client applications/users [8].

The WoT is intended to enable interoperability across IoT platforms and application domains, enabling the discovery of Things for interaction with other Things or applications. This type of Thing exposure facilitates the creation of mashups, where services/data from one or multiple Things are combined with virtual Web resources, e.g., to decide on an actuation [9]. In the context of smart cities and related systems, as more Things become available and mashups are built, more data will be collected, analyzed and processed, which brings storage and processing capability challenges. One way to solve this is to use cloud infrastructures, as proposed in [9,10]. However, cloud computing layers involve high delays, mainly due to the distance between the devices and cloud data centers. This can be avoided if developers push processing of sensed data to edge devices close to the physical Things, reducing latency and saving energy. Edge computing is the paradigm to use in order to achieve this goal, as it brings computing and data storage closer to where it is needed. Many emerging IoT services are expected to benefit from such a paradigm, since low latency and processing, e.g., for security purposes, will be critical requirements in many scenarios [11], e.g., emergency services [12]. Within mobile networks, this is handled by Multi-access Edge Computing (MEC) infrastructures that are expected to be incorporated into future 6G networks, creating the so-called distributed and collective edge intelligence [13,14]. Edge devices become intelligent hubs that are able to deliver highly personalized services directly from the edge of the network, enabling applications to perform at their best. These types of architectures strongly benefit from breaking monolithic applications into loosely coupled microservices for greater flexibility and robustness of IoT applications, improving their performance.

In this article, the Microservices-compliant Load Balancing (McLB) problem in MEC infrastructures is addressed, and a solution is proposed that joins complex network theory with multiplex dimension representation and analysis of these networks. This approach represents a change of perspective that allows multiplexity to capture the complexity of WoT mashup organizations, both in terms of their internal interdependencies and their impact on the cognitive organizational aspect of MEC devices/servers [15,16]. The aforementioned scenario is modeled using two interdependent and weighted multiplex networks [17–19]. The first refers to WoT mashups, while the second is populated by MEC servers whose computational load depends on the way in which Web resources are organized, used and, if possible, reused in order to provide the final mashup applications. The impact of the extracted knowledge is quantified through the use of heuristics that are engineered to guarantee load balancing and, consequently, Quality of Service (QoS).

The contributions of this article include:

- Modeling approach to address the McLB problem in MEC infrastructures. The approach allows for the unveiling of the interplay between WoT mashups (their use, reuse and chaining) and related computational loads at the MEC.
- Definition of a weight system for WoT resources, at one of the weighted multiplex networks, by taking into account the concept of input heterogeneity. How such a weight system influences the workload of the second multiplex network is also pointed out. For the second weighted multiplex network, an interaction weight system is defined that is based on computing capabilities of MEC server pairs.
- Evaluation of the interplay between computational loads at MEC servers and multiplex structural properties, through the definition of a complex involvement that takes into account a participation coefficient, the strength of links, and the reuse of WoT resources.
- Evaluation of time and energy overhead for both local computation and offloading to neighboring MEC servers, shedding light on the impact of complex involvement

measurements in triggering distributed cognitive-based decision mechanisms. This awareness makes it possible to reduce costs and improve QoS.

The remainder of this paper is organized as follows. In Section 2, some background and methods are discussed to highlight the importance of MEC, WoT and microservices-based deployments as key enabling factors for 6G. Multiplex networks and complex theory perspective in 6G are also introduced. Section 3 provides WoT and microservices-related definitions and assumptions, to clarify how smart city mashups would be deployed, and then dives into the analytical approach that was taken to model the McLB problem. Section 4 presents results and discussion, and Section 5 concludes the article and proposes future work.

2. Background and Methods

2.1. MEC for Smart City Applications

Cities are evolving into constantly changing systems of networked Things and people, and the introduction of MEC represents a pivotal point in developing innovative and cognitive urban applications [13,14]. Together with Artificial Intelligence (AI), the MEC is able to assure the intelligence required to process data accurately and efficiently, at the edge, creating a powerful distributed computing environment. Networks get self organizing and sustaining capabilities, in order to satisfy the dynamically changing services' requests and make more efficient use of resources. Typical 6G applications, such as self-driving cars, traffic and logistic management systems, telepresence and virtual, augmented or extended reality, require real-time feedback to be effective in addressing challenges from real-world scenarios. These aspects are particularly challenging and require efficient distributed models, data re-utilization and dynamic decision mechanisms based on all the knowledge available from different models and data sources [1,20]. For these reasons, the interest in modern interdisciplinary concepts, known as Edge Intelligence (EI), is growing and it is now a key pillar aspect of 6G [13].

Research activities are now shifting from traditional cloud-based architectures to collaborative, distributed, low-latency and reliable ones, calling for a shift from centralized/cloud-based systems to new EI-based designs. In the latter case, however, data are distributed across heterogeneous end devices, both Base Stations (BSs) and/or mobile devices (e.g., phones, cameras, vehicles, drones), characterized by limited computation and storage capabilities. Mobile end devices, for example, can be hand-held or wearable systems that are powered by capacity-limited batteries and storage. For these devices to stay cheap, it is crucial to rely on additional systems that acquire, cache, process and analyze data close to their source. These additional systems can form a distributed support network, shared by many systems, allowing for scalability, adaptability and resilience. Another crucial aspect is to make the best use of resources at the edge of the network, for the support of an entirely new generation of services. As an example, power consumption and latency issues are still open questions in MEC networks, and have a significant impact on the smart city application support.

In a nutshell, it is crucial to allow software to be “liquid” for continuous “flow” across devices and relocation of computation after design time, instead of making any kind of reservation “a priori”. Microservices are a software development methodology that goes in this direction [21,22]. Modular lightweight application components are individually deployed on-demand, contrary to monolithic software applications whose modules cannot be independently executed, becoming more suitable for distributed systems. Microservices are cohesive, autonomic, replaceable and deployable independent processes interacting with each other through standardized interfaces [23].

2.2. Multiplex Networks

Key concepts in complex systems theory, such as adaptability, resilience, decentralization, self-organization and self-optimization, become crucial for 6G communication networks, which are characterized by a high level of interdependence among their highly

heterogeneous components and by dense deployments. Complex systems analysis can be considered as the science that studies how elements of a system act collectively, to determine the emerging behavior of the whole system, and how they interact with the environment. For these reasons, the complex system approach represents an effective and useful tool to monitor, model and design the previously mentioned networks [2].

Complex systems are totally described by their connectedness: *elements having connections of a certain relationship type*. This means that network representations of complex systems include nodes that interact with each other through multiple links. A traditional monolayer approach can be used to describe the network, leading to the analysis of a single graph including all links. As this approach assumes a single kind of link, relevant information ends up being neglected and knowledge about the structural complexity and connectivity of the system is lost. In fact, in many real applications the relationships between nodes may differ in relevance, context and meaning [17–19]. In order to preserve and extract this knowledge, it is crucial to introduce a multi-dimensional network analysis, which allows an in-depth assessment (in structural terms) of the encoded information. Multilayer networks are a generalization of traditional graphs and provide a novel framework to model environments where nodes are connected differently in different layers [17,18]. That is, there will be intra-layer connections (within a layer) and inter-layer connections (between layers).

A multiplex network is a particular kind of multilayer network where the set of nodes is the same for all layers, and a node can be adjacent to another in the same layer, through intra-layer edges, or to its counterpart in another layer, through inter-layer edges [17,18,24]. Complexity in connections can be explored more deeply when weighted multiplex networks are used, where weights reflect intensity of interactions. On the basis of these premises, links between nodes can be distinguished not only by the kind of interaction but also by their intensity [19]. Therefore, multiplexity adds an additional dimension of analysis that allows non-trivial dynamics, phenomena and interdependencies of many real-world systems to be unveiled. More information and knowledge are extracted in order to understand a phenomena, to identify mechanisms and to inspire the design of such systems.

The relevance of multiplex network modeling is unquestionable and the scientific interest around applications is growing in several areas: biological, social and technological systems, social networks and telecommunication ones, epidemics and social contagions, transportation networks and brain computing dynamics [24–26]. Its application to smart cities, involving WoT devices supported by an MEC distributed infrastructure, allows for the engineering of heuristics that guarantee load balancing and, consequently, QoS to applications.

3. Modeling a Distributed MEC for WoT Mashups

3.1. Smart City Mashup Scenario

IoT cameras made available in a smart city can be used by visitors, to take videos/photos, while acting alongside other sensors (e.g., motion or smoke detectors) for security purposes or to ensure that safety regulations are followed. Other applications may include video analytics, location services, augmented reality, object recognition, etc. The devices will be shared by client applications building mashups, for use by locals or anyone around the world. Such mashups can integrate processing tasks to improve processes/services or to create new ones. MEC can create a powerful distributed computing environment that can be deployed to support low-latency services and IoT applications. Scenarios, such as industrial automation, smart homes, smart vehicles, among others, are envisioned as 6G use cases due to their strict latency and reliability requirements. Applications with coordination and contextualization needs, such as Intelligent Transportation and Logistics, are also expected to benefit from 6G.

3.2. Basic Definitions and Assumptions

The core idea of the WoT is that Things will be exposing services and data through RESTful APIs that other developers and devices can easily understand and use. REST is a resource-oriented architecture where every component of a system (sensor, sampling frequency, variable, etc.) is called a resource, which can be individually addressed using a Uniform Resource Identifier (URI) standard scheme [7]. Resources can also enforce processing or decision tasks [27].

Definition 1 (Task). Resource implementing autonomous processing or decision logic on certain inputs and returning an output value. Besides simple logic, complex processing (eventually integrating other info; e.g., historical data) can be performed. The overall set of different tasks is denoted by \mathcal{T} .

The exposure of Thing resources facilitates the creation of mashups. In general, a mashup is a way to compose a new service from existing services [28]. While this definition is focused mainly on information services, recent efforts on WoT standardization by W3C (see [8]) are allowing Thing resources to become part of mashups [9,29]. Recently, Rule resources have been introduced to implement observe–evaluate–actuate patterns, which are required in mashups [27].

Definition 2 (Rule Resource). Collection including a reference to a set of input resources, a task, and a set of output resources where task results should be placed. It is a REST-compliant mechanism to build observe–evaluate–actuate workflows. For a given rule \mathcal{R}_i , the set of inputs is denoted by $\mathcal{I}(\mathcal{R}_i)$, the set of outputs by $\mathcal{O}(\mathcal{R}_i)$, and the task by $t(\mathcal{R}_i) \in \mathcal{T}$ (see Figure 1).

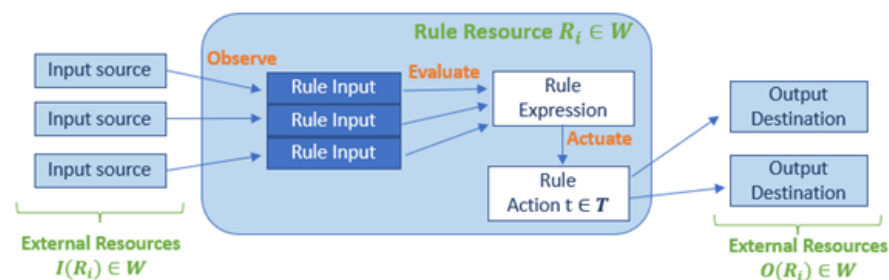


Figure 1. Rule resource composition.

Note that every element of the Rule collection (inputs, task, outputs) supports Create, Read, Update and Delete (CRUD) operations, which means that these can be modified on-the-fly (see [27] for more details). The task feeds on the set of Rule input resources, and this set can change. Each input resource can also go through some format conversion or special treatment. The set of Rule output resources can also change on-the-fly, and notification templates are used for format adaptation. Rule inputs and outputs are Web resources, $\mathcal{I}(\mathcal{R}_i) \subseteq \mathcal{W}$ and $\mathcal{O}(\mathcal{R}_i) \subseteq \mathcal{W}$, where \mathcal{W} denotes the overall set of Web resources, which means that Rule inputs, task and outputs can be hosted in different URIs and physical locations.

Definition 3 (Web Mashup). Workflow linking Web resources, which are identifiable digital, physical or abstract entities. A mashup \mathcal{S}_i can be seen as a set of Rule resources and, therefore, can be denoted by $\mathcal{S}_i = \{\mathcal{R}_1, \mathcal{R}_2, \dots, \mathcal{R}_{|\mathcal{S}_i|}\}$.

Note that a Rule resource can be part of multiple mashups because there can be multiple Rule outputs, each with its own format, for participation in different workflows.

In future MEC network architectures there will be authorized third parties, such as application developers and content providers, which will be able to use application servers integrated at the Radio Access Network (RAN), which causes a multi-tenancy run-time and

hosting environment for applications to emerge. Since MEC is a distributed cloud platform, breaking a monolithic application into loosely coupled microservices can bring significant gains in the performance, flexibility and robustness of IoT applications, as it is possible to relocate or replicate microservices. Microservices also have the advantage of being reusable across applications. Thus, an IoT related task can be implemented monolithically, or it can be broken down into microservices (e.g., data processing, log file or database update).

Definition 4 (Microservices). *Independently deployable services communicating through a well-defined lightweight mechanism. A set of microservices serves a certain business goal/task. The available microservices are denoted by \mathcal{V} , and the set of microservices involved in task $t \in \mathcal{T}$ is denoted by \mathcal{V}_t .*

Definition 5 (Distributed MEC). *Network solution providing services and computing functions required by clients at the edge. The set of application servers, or data centers, is denoted by \mathcal{D} . Each $d \in \mathcal{D}$ provides computing resources, storage capacity, connectivity and access to RAN information, and will be hosting microservices instances running as virtual appliances.*

In short, a microservices architecture can better ensure that applications are always on, due to replication, while a MEC infrastructure ensures a low end-to-end latency. This type of architecture, together with the uncoupling of Rule collection components, provides the key elements for edge computing to work properly. Mashups can be built independently by the client, using a Rule-like mechanism, and processing/service tasks inside Rules can be moved to the MEC. Naturally, the MEC would benefit from load balancing across application servers, which would have to take into account interdependencies among microservices, i.e., the load on a successor microservice instance depends on the load from predecessor microservice instances [30].

Load balancing becomes the challenge that needs to be solved and aspects such as MEC virtualization facilities and interdependencies among microservices have to be considered. This problem is defined as follows.

Definition 6 (Microservices-compliant Load Balancing (McLB) problem). *Given a distributed MEC serving a set of client mashups, whose tasks involve one or more microservice instances hosted at MEC servers, plan for an adequate cooperative server MEC environment where load balancing is ensured.*

In this article, multiplex network modeling is used to address this problem, as explained next.

3.3. Multiplex Network Modeling for the McLB Problem

3.3.1. General Formalization of Multiplex Networks

In general, a multiplex network \mathcal{M} is defined as a set of L networks representing the different layers. Each network is denoted by the graph $\mathcal{G}_\alpha = (\mathcal{N}, \mathcal{E}_\alpha)$, where \mathcal{N} is the set of nodes/vertices, the same in every layer, and \mathcal{E}_α is the set of links/edges, which changes according to the layer $\alpha \in \{1, \dots, L\}$ [17,19].

A network \mathcal{G}_α can be described by its adjacency matrix denoted by \mathbf{A}^α , where $a_{ij}^\alpha = 1$ if there is a link from i to j , and $a_{ij}^\alpha = 0$ otherwise. In weighted multiplex networks, a link ij in \mathcal{E}_α will be characterized by the weight w_{ij}^α . This means that $a_{ij}^\alpha = w_{ij}^\alpha$ whenever there is a weighted link between i and j , and $a_{ij}^\alpha = 0$ otherwise [19].

Weighted multiplex networks allow node properties to be defined: (i) k_i^α , for degree; (ii) s_i^α , for strength; and (iii) o_i , for the overlapping degree; where i is a node in layer α . These are defined as follows [17,19] (see Figure 2):

$$k_i^\alpha = \sum_{\{j \in \mathcal{N}: j \neq i\}} a_{ij}^\alpha \quad (1)$$

$$s_i^\alpha = \sum_{\{j \in \mathcal{N} : j \neq i\}} w_{ij}^\alpha \quad (2)$$

$$o_i = \sum_{\alpha=1}^L k_i^\alpha \quad (3)$$

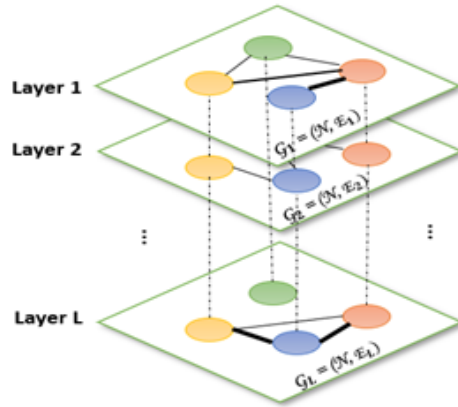


Figure 2. Multiplex network representation.

The Z-score is a measure associated with o_i , and is calculated as:

$$z(o_i) = \frac{o_i - \langle o \rangle}{\sigma_o} \quad (4)$$

where $\langle o \rangle$ is the average overlapping degree of the nodes in the system, and σ_o is the corresponding standard deviation [31]. The examination of Z-score variation among nodes is crucial to introduce a classification of nodes that is based on the quantification of multiplex network properties. Another information of interest is the heterogeneity of the number of neighbors across layers, called the multiplex participation coefficient [17] for node i , which is measured using:

$$P_i = \frac{L}{L-1} \left[1 - \sum_{\alpha=1}^L \left(\frac{k_i^\alpha}{o_i} \right)^2 \right] \quad (5)$$

where $P_i = 1$ when the links incident on node i are equally distributed across the layers, and $P_i = 0$ when the node is active on only one layer.

Taking into account the mentioned multiplex network properties, two interdependent weighted multiplex networks are proposed to model the McLB problem: \mathcal{M}_1 , denoting a multiplex network that models the WoT section; and \mathcal{M}_2 , denoting a multiplex network that models the MEC section. These networks are shown in Figure 3, and their role is clarified next.

3.3.2. WoT Multiplex Network

The multiplex network \mathcal{M}_1 is intended for WoT section modeling, where client mashups are built. Therefore, its nodes represent Rule resources exposed by Things, while its links/edges represent workflow wiring together such Rule resources. Rule resources can be part of multiple mashups, and \mathcal{M}_1 layers are the reflection of these structural interconnections. Thus, for a set of up and running mashups, the network graph at layer β , for \mathcal{M}_1 , is denoted by $\mathcal{G}_\beta^{\text{WoT}} = (\mathcal{R}^{\text{WoT}}, \mathcal{E}_\beta^{\text{WoT}})$, where $\mathcal{R}^{\text{WoT}} = \cup_{\beta \in \{0, \dots, L^{\text{WoT}}\}} \mathcal{S}_\beta$. The set \mathcal{S}_β includes the Rule resources used by the mashup defined at layer β . Additionally, $\mathcal{G}_\beta^{\text{WoT}} \equiv \mathcal{G}_\alpha$, $L^{\text{WoT}} \equiv L$, $\mathcal{R}^{\text{WoT}} \equiv \mathcal{N}$ and $\mathcal{E}_\beta^{\text{WoT}} \equiv \mathcal{E}_\alpha$.

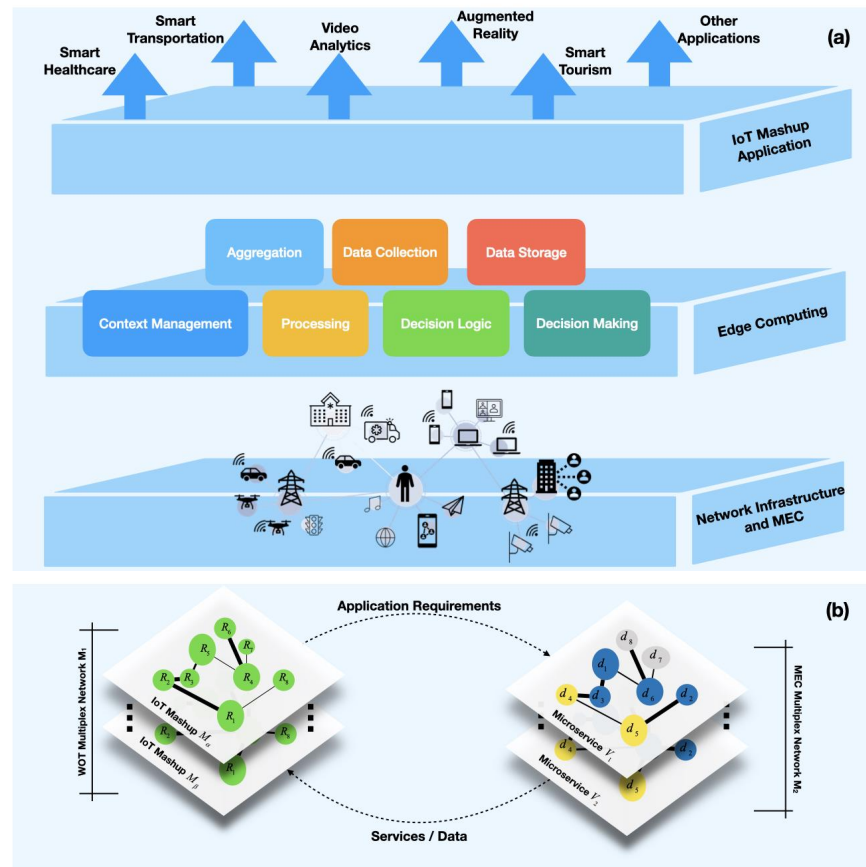


Figure 3. Multiplex network interdependency model (represented in (b)) for the McLB problem (synthesized in (a)).

A Rule resource \mathcal{R}_i is a Web resource of a particular type, available at some URI. As stated in Definition 3, the task performed by a Rule uses input Web resources, denoted by $\mathcal{I}(\mathcal{R}_i)$, and after performing some task the output is placed at one or more output URIs, $\mathcal{O}(\mathcal{R}_i)$, which will feed into other Rules. According to [27], the overall execution logic of the Rule resource is the following: *every time a Rule input is updated, the Rule task is executed and if its state changes from False to True then notifications are sent to the output resources*. When abstracting ourselves from intermediate URIs, it is possible to state that \mathcal{R}_i task execution is triggered by Rules that precede Rule \mathcal{R}_i in the workflow, within the same layer or in other layers, and this influence will be modeled as follows. For a weighted multiplex network \mathcal{M}_1 , it is assumed that the relevance of a link is greater when its endpoints have greater difference on the number of inputs. The weights of \mathcal{M}_1 interactions will be the following:

Definition 7 (Interaction weights in \mathcal{M}_1). *In a weighted multiplex network \mathcal{M}_1 , the weight of a link between endpoints \mathcal{R}_i and \mathcal{R}_j in layer α , denoted by $w_{\mathcal{R}_i, \mathcal{R}_j}^\alpha$, is given by:*

$$(w_{\mathcal{R}_i, \mathcal{R}_j}^\alpha)^{\mathcal{M}_1} = 1 + |(k_{\mathcal{R}_i}^\alpha - 1) - (k_{\mathcal{R}_j}^\alpha - 1)| \quad (6)$$

where $k_{\mathcal{R}_i}^\alpha$ and $k_{\mathcal{R}_j}^\alpha$ are the degrees of the \mathcal{R}_i and \mathcal{R}_j nodes in layer α , respectively.

The degree of a node is the number of interactions in layer α . Figure 4 illustrates such intra-layer interaction weights. This property is useful to shape interactions among nodes in a layer, allowing re-usability and chaining of resources to be improved. In addition, Rule resources interact with their counterparts in other layers. This is so because Rule resources are not only chained for the creation of a specific mashup, but also contribute

to the formation of other mashups. This is captured by the interdependent layers of the multiplex network \mathcal{M}_2 .

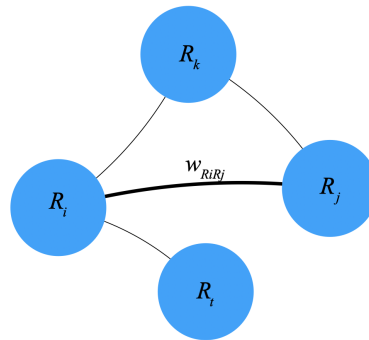


Figure 4. Weights of interactions in \mathcal{M}_1 .

3.3.3. MEC Multiplex Network

The multiplex network \mathcal{M}_2 is intended for MEC section modeling, where application servers or data centers are placed. While the network is populated by the set of application servers/data centers (network nodes), the layers refer to the microservices. In each layer, an $n \in \mathcal{N}$ will be hosting microservices instances running as virtual appliances. Multiplex representations allow us to establish relationships and to shed light on complex interdependence between microservices with respect to load distribution, reusability, computing capability of hosting nodes and time involved. Therefore, links between nodes in \mathcal{M}_2 will have weights defined as follows.

Definition 8 (Interaction weights in \mathcal{M}_2). Referring to the weighted multiplex network \mathcal{M}_2 , the weight of the link w_{ij}^α is given by:

$$(w_{ij}^\alpha)^{\mathcal{M}_2} = 1 + |f_i^\alpha - f_j^\alpha| \quad (7)$$

where f_i^α and f_j^α represent the computing capability of data centers i and j (CPU/cycles per second [32]), respectively, which is dedicated to the microservice α . This accounts for multiple microservice instances.

3.3.4. WoT and MEC Interplay

By assuming that WoT applications can share Rules, when building their workflows, and that Rule tasks can be decomposed into small and light components/microservices, it becomes possible to distribute microservices across servers. This leads to complex interactions between MEC servers and, given the chains of resource requirements involved, multiple levels of abstractions with complex interactions can emerge. To capture the connection between Rule resources in \mathcal{M}_1 and microservices distributed across servers populating \mathcal{M}_2 , the complex load for each node i in \mathcal{M}_2 is defined. This represents the computation overhead due to the involvement of MEC nodes in the multiplex network representation. The complexity measure $(b_i)^{\mathcal{M}_2}$ is defined as follows:

Definition 9 (Complex involvement of server i in \mathcal{M}_2).

$$(b_i)^{\mathcal{M}_2} = (P_i)^{\mathcal{M}_2} \times \sum_{\alpha=1}^{L_2} (s_i^\alpha)^{\mathcal{M}_2} \times \frac{\sum_{\beta=1}^{L_1} \sum_{\{j \in \mathcal{N}_1\}} (s_j^\beta)^{\mathcal{M}_1}}{\sum_{\{j \in \mathcal{N}_1\}} (P_j)^{\mathcal{M}_1}} \quad (8)$$

where L_1 and L_2 denote the number of layers in \mathcal{M}_1 and \mathcal{M}_2 , respectively, while \mathcal{N}_1 denotes the set of nodes in \mathcal{M}_1 .

Note that $(P_i)^{\mathcal{M}_2}$ in Equation (8) is the multiplex participation coefficient of a node $i \in \mathcal{N}_2$, a set of nodes in \mathcal{M}_2 , measuring the involvement of MEC servers' resources in different microservices/layers. The higher the participation coefficient, the higher the computation load at the server, due to the fact that the node actively participates in multiple microservices. The $\sum_{\alpha=1}^{L_2} (s_i^\alpha)^{\mathcal{M}_2}$ is the sum of the strengths of node i , calculated for each layer $\alpha \in \{1, \dots, L_2\}$. This measure depends on the importance of links, which has to do with the computing capability difference of its nodes. The last component of Equation (8) is the Reuse function for Rule resources in \mathcal{M}_1 , where $\sum_{\beta=1}^{L_1} \sum_{j \in \mathcal{N}_1} (s_j^\beta)^{\mathcal{M}_1}$, summing the strengths of nodes in \mathcal{M}_1 over all layers $\beta \in \{1, \dots, L_1\}$, takes into account the heterogeneity of links as a consequence of \mathcal{M}_1 's weights (see Section 3.3.2). A greater strength implies greater heterogeneity at Rule resource inputs and, consequently, additional loading for MEC servers in \mathcal{M}_2 . The $(P_j)^{\mathcal{M}_1}$ is the multiplex participation coefficient of node $j \in \mathcal{N}_1$, from the WoT-based multiplex network \mathcal{M}_1 , and it represents a measure of how Rule resources in \mathcal{M}_1 are reused and chained in different mashups. The greater the multiplex participation coefficient, the more Rules are involved in different mashups, with no additional computation required on MEC servers. The $(b_i)^{\mathcal{M}_2}$ ends up measuring the commitment of MEC node i and, therefore, the probability of opting for local computation or offloading to neighboring nodes. Thanks to this complex measure, an estimation is made regarding the probability that a node in \mathcal{M}_2 will compute locally or not.

3.3.5. Impact on Energy and Time

To measure the impact on both energy and time associated with computing that is held locally and offloaded, a cost function must be defined. As specified in Definition 4, each task $t \in \mathcal{T}$ can be served by a set of microservices \mathcal{V}_t and, for the sake of simplicity, it is assumed here that such a data load is initially distributed across microservices and MEC nodes (layers and nodes in \mathcal{M}_2 , respectively) using a hypothetical distribution that depends on the degree k_i^α .

Let $x_{it} \in \{0, 1\}$ denote the offloading decision for node i regarding the task $t \in \mathcal{T}$. That is, $x_{it} = 0$ if the MEC server goes for the local computation, and $x_{it} = 1$ in case of offloading. According to such a decision, the energy and time cost is defined as follows.

Definition 10 (Decision Impact on Energy and Time). *The total cost associated with decisions stored in matrix $\mathbf{X}_{|\mathcal{N}_1| \times |\mathcal{T}|}$ is defined by:*

$$\text{Cost}(\mathbf{X}) = \sum_{i=1}^{|\mathcal{N}_1|} \sum_{t=1}^{|\mathcal{T}|} (E_{it} + H_{it}) \quad (9)$$

where E_{it} is the required energy and H_{it} the execution time for task t initially allocated at node i . As operations will be performed locally or offloaded:

$$E_{it} = (1 - x_{it})E_{it}^{\text{loc}} + x_{it} \times E_{it}^{\text{off}} \quad (10)$$

$$H_{it} = (1 - x_{it})H_{it}^{\text{loc}} + x_{it} \times H_{it}^{\text{off}} \quad (11)$$

whose components are defined as follows [32]:

$$H_{it}^{\text{loc}} = \frac{c_{it} \times l_{it}}{f_i}, \quad (12)$$

where c_{it} is the number of CPU cycles, l_{it} is the task load and f_i is the computing capability of node i in \mathcal{M}_2 ,

$$H_{it}^{\text{off}} = \frac{l_{it}}{r_i} + \frac{c_j \times l_{jt}}{f_j} \quad (13)$$

where f_j refers to the computing capability of node j in \mathcal{M}_2 , to which the task load is offloaded, and $r_i = B \times \ln(1 + \frac{p_i}{\omega_0 B})$ is the uplink rate for bandwidth B , p_i is the transmission power, and ω_0 is the channel noise,

$$E_{it}^{loc} = v_i \times l_{i,t} \times c_{i,t} \quad (14)$$

with v_i energy per CPU cycle and, finally,

$$E_{it}^{off} = p_i \times H_{it}^{off}. \quad (15)$$

4. Analysis of Results

4.1. Scenario Setup

Simulations were conducted considering a multiplex network \mathcal{M}_1 with $L_1 = 3$ layers, with each layer representing an IoT mashup, which models three distinct kinds of weighted interactions and connectivity among the $|\mathcal{N}_1| = 1000$ nodes (Rule resources). A variable population for \mathcal{M}_2 , $50 \leq |\mathcal{N}_2| \leq 500$ nodes (MEC servers), and a variable number for L_2 (microservices layers), $3 \leq L_2 \leq 9$, are assumed. Each layer of both weighted multiplex networks has a Scale-Free (SF) network topology [33]. SF networks are highly heterogeneous networks, characterized by a power-law degree distribution, with a high degree of correlation between nodes and degree distribution having a long tail. SF networks fit many real-world networks and are characterized by preferential attachment and growth; new nodes are added to the existing ones with a probability of attachment that is proportional to the degree of older nodes in the network.

Regarding the remaining simulation parameters, it is assumed that the size/load of tasks is in the range $[0, 20]$, i.e., $l_{it} \in [0, 20]$ MB, and that tasks are spread across nodes following the hypothetical distribution described in Section 3.3.5. The energy consumption per CPU cycle is set to $v_i = (0.20 \times 10^{(11)})$ Joules per cycle. For the number of CPU cycles it is considered that there will be $c_{it} = 500$ cycles per bit. In order to take into account the heterogeneous computing capability of servers, a random computing capability, $f_i \in \{0.5, 0.6, \dots, 1\}$ GHz, is used. The device's transmission power, channel bandwidth and background noise are $p_i = 0.1$ W, $B = 20$ MHz and $\omega_0 = -100$ dBm, respectively [32].

To build the model, perform computation and obtain results, the programming language R and IDE RStudio were used. Figures were generated using the Plotly and ggplot2 packages [34–36]. Simulations ran on a personal computer with Intel(R) Core(TM) i7-8750H CPU, 8 GB RAM capacity and 2.20GHz frequency. The employed simulation settings are summarized in Table 1.

Table 1. Synthesis of simulation conditions, tools and packages.

Number of nodes in \mathcal{M}_1	$ \mathcal{N}_1 = 1000$
Number of layers in \mathcal{M}_1	$L_1 = 3$
Number of nodes in \mathcal{M}_2	$50 \leq \mathcal{N}_2 \leq 500$
Number of layers in \mathcal{M}_2	$3 \leq L_2 \leq 9$
Size/load of tasks	$l_{it} \in [0, 20]$ MB
Energy consumption per CPU cycle	$v_i = (0.20 \times 10^{(11)})$ Joules per cycle
CPU cycles per bit	$c_{it} = 500$ cycles per bit
Channel bandwidth	$B = 20$ MHz
Computing capability	$f_i \in \{0.5, 0.6, \dots, 1\}$ GHz
Transmission power	$p_i = 0.1$ W
Background noise	$\omega_0 = -100$ dBm
Tools	R, RStudio [36]
Packages	Plotly and ggplot2 [34,35]

4.2. Evaluation

Figure 5a shows a cartography of the node roles in the multiplex network \mathcal{M}_2 , plotting for each node i the multiplex participation coefficient $(P_i)^{\mathcal{M}_2}$ versus the Z-score $(z(o_i))^{\mathcal{M}_2}$ of the total overlapping degree. Since the overlapping degree of a node represents its overall importance in terms of number of incident edges, while the multiplex participation coefficient gives information about the distribution of incident edges across the layers, this cartography allows us to classify nodes merely in terms of their structural role in the multiplex network (see [17,31]).

In general, representing nodes as points in the $P_i - z(o_i)$ plane allows us to identify three classes of nodes: (i) focused, comprising the nodes for which $0 \leq P_i \leq \frac{1}{3}$; (ii) mixed, comprising nodes having $\frac{1}{3} \leq P_i \leq \frac{2}{3}$; (iii) multiplex nodes, comprising nodes with $P_i > \frac{2}{3}$. Regarding the Z-score of their overlapping degree, it is possible to distinguish: (i) hubs, for which $z(o_i) \geq 2$; (ii) regular nodes, for which $z(o_i) < 2$. Moreover, through the color of each node it is possible to show the variation of the mean strength distribution $\langle s_i \rangle^{\mathcal{M}_2}$ and through its size, the complex involvement $(b_i)^{\mathcal{M}_2}$ is shown.

A finding from such a cartography is that greater values of complex involvement $(b_i)^{\mathcal{M}_2}$ can be noticed in nodes classified as regular multiplex, confirming that the multiplex network representation is quite suitable to capture the complexity of the considered scenario, where computation is shifted to the edge. These nodes are regular because they have a low number of incident edges and multiplex as their links are well distributed across the layers (microservices in which they are involved). This means these nodes are involved in many microservices but their computation is not crucial for any of them. Furthermore, we can deepen the analysis, evaluating not only the number of nodes' incident links and their distribution across the layers but also the distribution of their average strengths. The strength is a function of weights (see Equation (2)) and according to Definition 8 it depends on the difference of computing capabilities among linked MEC nodes. A poor heterogeneity in the mean strengths' distribution $\langle s_i \rangle^{\mathcal{M}_2}$, as evidenced from the Figure 5a, is consistent with the scenario in consideration in which the nodes at the edges, which carry out the computation, have limited and not too different computing capabilities.

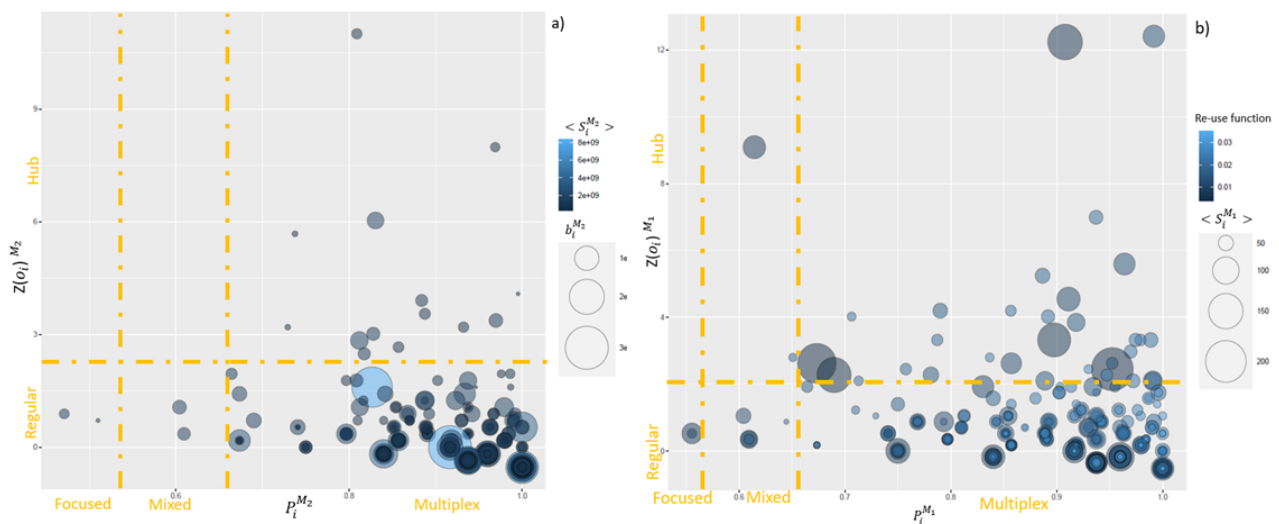


Figure 5. Role of nodes: (a) multiplex participation coefficient $(P_i)^{\mathcal{M}_2}$ versus the Z-score $(z(o_i))^{\mathcal{M}_2}$, for each node in \mathcal{M}_2 (color indicates the average strength $\langle s_i \rangle^{\mathcal{M}_2}$ and the size of nodes represents the measure of complex involvement $(b_i)^{\mathcal{M}_2}$); (b) multiplex participation coefficient $(P_i)^{\mathcal{M}_1}$ versus Z-score for each node in \mathcal{M}_1 (color indicates the Reuse function of resources in \mathcal{M}_1 and the size of nodes represents the average strength $\langle s_i \rangle^{\mathcal{M}_1}$).

The complex involvement $(b_i)^{\mathcal{M}_2}$ represents the interplay between the two multiplex networks \mathcal{M}_1 and \mathcal{M}_2 (see Section 3.3.4) and strictly depends on the way in which the

resources are organized and reused in the Wot-based \mathcal{M}_1 , represented through the Reuse function. For this reason, Figure 5b shows the cartography in the plane $P_i - z(o_i)$ for the nodes in \mathcal{M}_1 . This time, the size represents the distribution of the average strength $\langle s_i \rangle^{\mathcal{M}_1}$, while the color is the variation of the Reuse function. This cartography sheds light on how the multiplex network analysis results in a deepened understanding of the hidden organization of Web resources, their use and re-use in IoT mashups, which impacts the computational load of MEC nodes in \mathcal{M}_2 . In this case, considering an SF as the network topology, it is possible to see that the most used and re-used Web resources are those classified as multiplex regular nodes. These Web resources participate in several IoT mashups but do not present a high number of incident links in any of them. The nodes with lower values of the Reuse function are the most critical nodes in terms of computation because they are characterized by high values of both $(P_i)^{\mathcal{M}_1}$ and $(z(o_i))^{\mathcal{M}_1}$ (multiplex hubs). This means that they are involved in many IoT mashups and, at the same time, in some of them they are important in terms of the number of incident links. In addition, those with high average strength are even more critical because they introduce further heterogeneity in computation (see Definition 7).

Figure 6a shows the cost for the case of $|\mathcal{N}_2| = 300$ nodes and $L_2 = 3$ layers. Points in the graph represent the cost (energy and time) associated with nodes for the following three cases: (i) nodes always choose local computation (orange points in the plot); (ii) nodes always decide on offloading (pink points in the plot); (iii) the decision about computation is made in accordance with the values of the complex involvement $(b_i)^{\mathcal{M}_2}$ (referred to as the cognitive case; green points in the plot). In order to provide a heuristic (non-optimal solution) to the McLB problem in a distributed and collective way, the mean values of costs related to the three cases are shown. From such information it is possible to state that the cognitive case represents the most profitable approach for the whole system, presenting the lowest mean cost. Figure 6b indicates the trend of the computing time, and it is clear that, in terms of time, the complex involvement choice is also quite convenient. This approach can be particularly suited to those distributed and mobile network contexts whose services require priority and are extremely sensitive to delay.

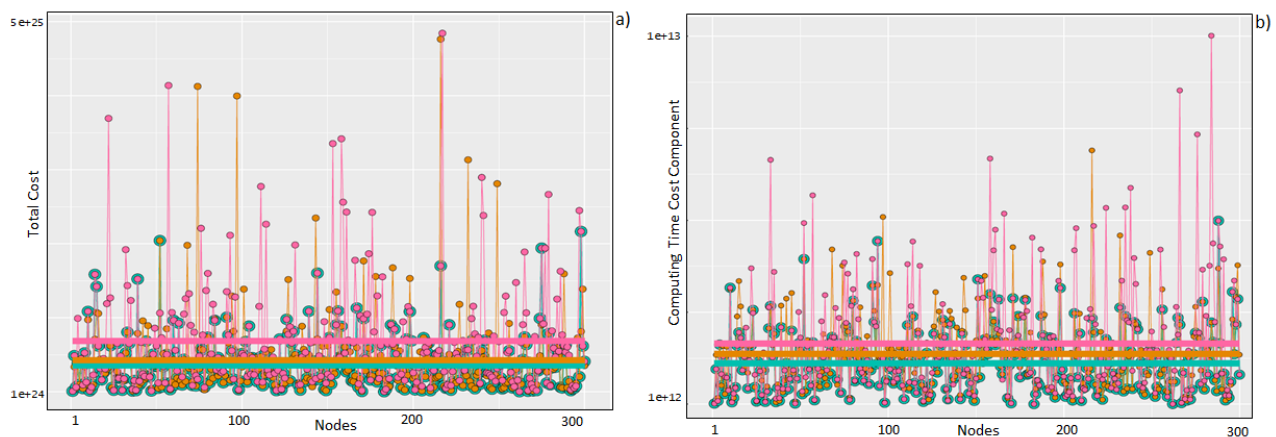


Figure 6. Cost H_{it} versus nodes: (a) total cost; (b) computing time component. Punctual and mean values are shown for three cases: local computation (in pink), offloading to neighbors (in orange), and cognitive computing decision (in green).

The mean and standard deviations, of both computing time and total cost, are reported in Table 2 for the following three cases: local computation, offloading to neighbors and cognitive computing decision.

Table 2. Significance of results.

	Local Computation	Offloading	Cognitive Computing
Mean Cost	7.15×10^{24}	5.33×10^{24}	3.90×10^{24}
Mean Execution Time	1.72×10^{24}	1.58×10^{12}	1.25×10^{24}
Standard Deviation of Cost	7.04×10^{24}	7.55×10^{24}	3.66×10^{24}
Standard Deviation of Execution Time	1.58×10^{12}	1.20×10^{12}	885,859,776,046

The heatmap in Figure 7 displays a comparison of the total cost for the local, offloading or cognitive computation cases. Different population sizes are used for the multiplex network \mathcal{M}_2 , $|\mathcal{N}_2|$ ranging from 50 to 500 nodes. The number of layers, L_2 , also varies and $L_2 = 3$, $L_2 = 5$, $L_2 = 7$ and $L_2 = 9$ cases are considered. The figure shows that the case where a computation decision is taken according to the knowledge extracted from the multiplex network representation, through the measure $(b_i)^{\mathcal{M}_2}$, the cognitive case presents the most profitable values for the whole system. Furthermore, as the population and the number of layers grow, this approach becomes increasingly convenient. In particular, for $L_2 = 9$ it is possible to see that a high number of layers means that the MEC nodes in \mathcal{M}_2 are involved in a large number of microservices, resulting in a major computation load and cost. Increasing the population, in the cognitive case, provides more capacity for self-organization, increasing the probability to go for the most profitable choice in terms of cost. In other words, a richer multiplex structure in terms of layers and nodes, even if it initially involves a higher computational load and cost, allows the exploitation of more knowledge, which results in a significant cost reduction.

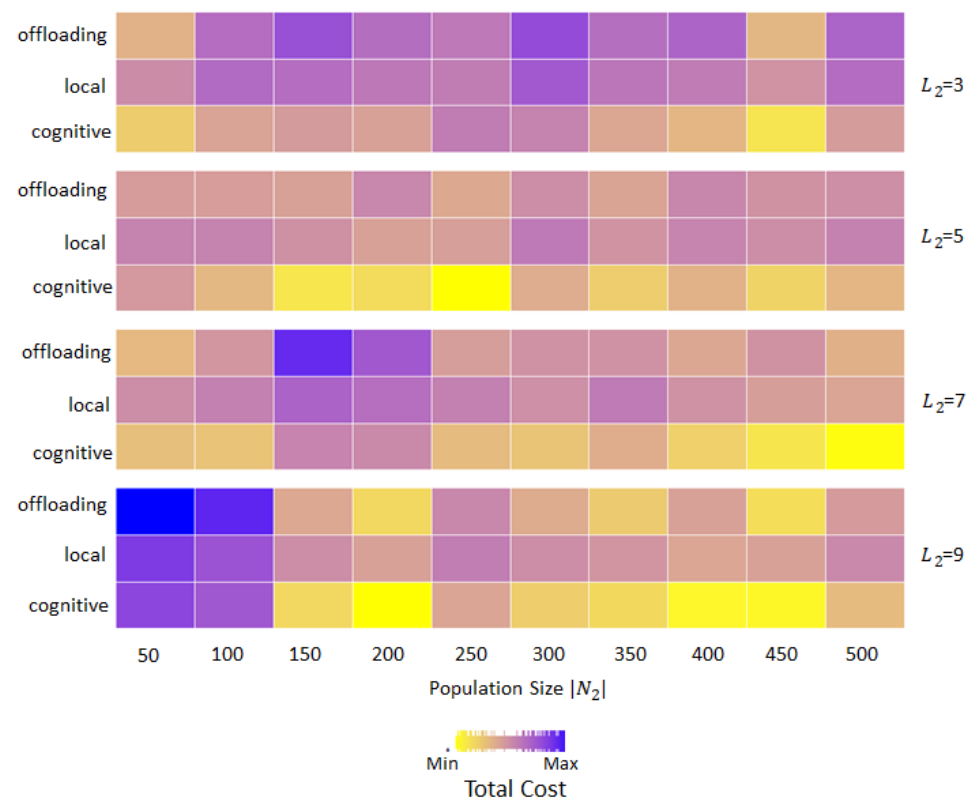


Figure 7. Heatmap of total cost sensitivity. For each computing decision considered (offloading to neighbors, local computation, cognitive computing), the heatmap is displayed considering a population size $|\mathcal{N}_2|$ basis (ranging from 50 to 500) and different numbers of layers ($L_2 = 3$, $L_2 = 5$, $L_2 = 7$ and $L_2 = 9$), where colors (ranging from yellow to blue) indicate the total cost values.

5. Conclusions and Future Work

This work presents an innovative multiplex networks-based approach to solve the Microservices-compliant Load Balancing (McLB) problem in MEC infrastructures. The novelty of the approach consists in introducing key features able to fully describe and capture the complexity of 6G ecosystems. These ecosystems involve WoT platforms characterized by heterogeneous Web resources and MEC servers devoted to computation. The aim is to achieve a connected intelligence from the network design, and control service construction and QoS, through the introduction of novel algorithms able to dynamically manage applications.

The results put in evidence that the multiplex network representation provides a suitable tool to capture the complex interactions and interdependence within MEC and WoT mashups, highlighting critical issues in Web resource usage and triggering cognitive decision mechanisms about the computation, with an impact on final services' QoS.

As future work, we envisage complex scenarios including also learning aspects and collective/evolutionary dynamics, such as cooperation through Evolutionary Game Theory, and mesoscopic analysis such as multilink communities detection, with the aim of improving performance in provided services.

Author Contributions: All the authors have equally contributed to the conceptualization, methodology, software, validation, formal analysis, investigation, resources, data curation, writing—original draft preparation and funding acquisition. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: This work was supported by FCT (Foundation for Science and Technology) from Portugal within CEOT (Center for Electronic, Optoelectronic and Telecommunications) and the UID/MULTI/00631/2020 project and by the Research Grants: Italian Ministry of University and Research (MIUR)—PON REC 2014–2020 within the project ARS01_01116 “TALISMAN” and by PO FSE 2014/2020-Sicilia 2020-Avviso 527 24/2018.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Tomkos, I.; Klonidis, D.; Pikasis, E.; Theodoridis, S. Toward the 6G network era: Opportunities and challenges. *IT Prof.* **2020**, *22*, 34–38. [\[CrossRef\]](#)
2. Sergiou, C.; Lestas, M.; Antoniou, P.; Liaskos, C.; Pitsillides, A. Complex systems: A communication networks perspective towards 6G. *IEEE Access* **2020**, *8*, 89007–89030. [\[CrossRef\]](#)
3. Scatà, M.; Di Stefano, A.; Giacchi, E.; La Corte, A.; Liò, P. The bio-inspired and social evolution of node and data in a multilayer network. In Proceedings of the 2014 5th International Conference on Data Communication Networking (DCNET), Vienna, Austria, 28–30 August 2014; pp. 1–6.
4. La Corte, A.; Marialisa, S. A process approach to manage the security of the communication systems with risk analysis based on epidemiological model. In Proceedings of the 2010 Fifth International Conference on Systems and Networks Communications, Nice, France, 22–27 August 2010.
5. La, C.; Aurelio, M.; Evelina, G. A bio-inspired approach for risk analysis of ict systems. In Proceedings of the International Conference on Computational Science and Its Applications, Santander, Spain, 20–23 June 2011.
6. Di Stefano, A.; Scatà, M.; Vijayakumar, S.; Angione, C.; La Corte, A.; Liò, P. Social dynamics modeling of chrono-nutrition. *PLoS Comput. Biol.* **2019**, *15*, e1006714. [\[CrossRef\]](#)
7. Guinard, D.; Trifa, V. *Building the Web of Things*; Manning Publications: Shelter Island, NY, USA, 2016.
8. W3C: Web of Things (WoT) Architecture. Available online: <https://w3c.github.io/wot-architecture/> (accessed on 30 November 2021).
9. Guerreiro, J.; Rodrigues, L.; Correia, N. Allocation of Resources in SaaS Clouds Managing Thing Mashups. *IEEE Trans. Netw. Serv. Manag.* **2020**, *17*, 1597–1609. [\[CrossRef\]](#)
10. Distefano, S.; Merlino, G.; Puliafito, A. A Utility Paradigm for IoT: The Sensing Cloud. *Pervasive Mob. Comput.* **2015**, *20*, 127–144. [\[CrossRef\]](#)
11. Casadei, R.; Fortino, G.; Pianini, D.; Russo, W.; Savaglio, C.; Viroli, M. A development approach for collective opportunistic Edge-of-Things services. *Inf. Sci.* **2019**, *498*, 154–169. [\[CrossRef\]](#)
12. Scatà, M.; Attanasio, B.; Aiosa, G.V.; La Corte, A. The dynamical interplay of collective attention, awareness and epidemics spreading in the multiplex social networks during COVID-19. *IEEE Access* **2020**, *8*, 189203–189223. [\[CrossRef\]](#)

13. Peltonen, E.; Bennis, M.; Capobianco, M.; Debbah, M.; Ding, A.; Gil-Castiñeira, F.; Yang, T. 6G white paper on edge intelligence. *arXiv* **2020**, arXiv:2004.14850.
14. Filali, A.; Abouaomar, A.; Cherkaoui, S.; Kobbane, A.; Guizani, M. Multi-access edge computing: A survey. *IEEE Access* **2020**, *8*, 197017–197046. [[CrossRef](#)]
15. Giacchi, E.; Di Stefano, A.; La Corte, A.; Scatà, M. A dynamic context-aware multiple criteria decision making model in social networks. In Proceedings of the International Conference on Information Society (i-Society 2014), London, UK, 10–12 November 2014; pp. 157–162.
16. Di Stefano, A.; Scatà, M.; La Corte, A.; Das, S.K.; Liò, P. Improving qoe in multi-layer social sensing: A cognitive architecture and game theoretic model. In Proceedings of the Fourth International Workshop on Social Sensing, Montreal, QC, Canada, 15 April 2019; pp. 18–23.
17. Battiston, F.; Nicosia, V.; Latora, V. The new challenges of multiplex networks: Measures and models. *Eur. Phys. J. Spec. Top.* **2017**, *226*, 401–416. [[CrossRef](#)]
18. Boccaletti, S.; Bianconi, G.; Criado, R.; Del Genio, C.I.; Gómez-Gardenes, J.; Romance, M.; Zanin, M. The structure and dynamics of multilayer networks. *Phys. Rep.* **2014**, *544*, 1–122. [[CrossRef](#)] [[PubMed](#)]
19. Menichetti, G.; Remondini, D.; Panzarasa, P.; Mondragón, R.J.; Bianconi, G. Weighted multiplex networks. *PLoS ONE* **2014**, *9*, e97857. [[CrossRef](#)] [[PubMed](#)]
20. Huynh, L.N.; Huh, E.N. Envisioning Edge Computing in Future 6G Wireless Networks. In Proceedings of the 2021 Fifth World Conference on Smart Trends in Systems Security and Sustainability (WorldS4), London, UK, 29–30 July 2021.
21. du Plessis, S.; Correia, N. A Comparative Study of Software Architectures in Constrained Device IoT Deployments. In Proceedings of the IEEE International Conference on Internet of Things and Intelligence System (IoTaIS), Bali, Indonesia, 1–3 November 2021.
22. du Plessis, S.; Mendes, B.; Correia, N. A Comparative Study of Microservices Frameworks in IoT Deployments. In Proceedings of the International Young Engineers Forum on Electrical and Computer Engineering (YEF-ECE), Costa da Caparica, Portugal, 3 July 2021.
23. Bucchiarone, A.; Dragoni, N.; Dustdar, S.; Lago, P.; Mazzara, M.; Rivera, V.; Sadovykh, A. Microservices. In *Science and Engineering*; Springer: Berlin, Germany, 2020.
24. Aleta, A.; Meloni, S.; Moreno, Y. A multilayer perspective for the analysis of urban transportation systems. *Sci. Rep.* **2017**, *7*, 1–9.
25. Santoro, A.; Nicosia, V. Algorithmic complexity of multiplex networks. *Phys. Rev. X* **2020**, *10*, 021069.
26. Orozco, L.G.N.; Battiston, F.; Iñiguez, G.; Szell, M. Extracting the Multimodal Fingerprint of Urban Transportation Networks. *Findings* **2020**, 13171.
27. Mazayev, A.; Correia, N. A Distributed CoRE-based Resource Synchronization Mechanism. *IEEE Internet Things J.* **2020**, *7*, 4625–4640. [[CrossRef](#)]
28. Cheng, B.; Zhao, S.; Qian, J.; Zhai, Z.; Chen, J. Lightweight Service Mashup Middleware With REST Style Architecture for IoT Applications. *IEEE Trans. Netw. Serv. Manag.* **2018**, *15*, 1063–1075. [[CrossRef](#)]
29. Guerreiro, J.; Rodrigues, L.; Correia, N. Resource Allocation Model for Sensor Clouds under the Sensing as a Service Paradigm. *Computers* **2019**, *8*, 18. [[CrossRef](#)]
30. Yu, R.; Kilari, V.T.; Xue, G.; Yang, D. Load Balancing for Interdependent IoT Microservices. In Proceedings of the IEEE INFOCOM 2019-IEEE Conference on Computer Communications, Paris, France, 29 April–2 May 2019.
31. Battiston, F.; Nicosia, V.; Latora, V. Structural measures for multiplex networks. *Phys. Rev. E* **2014**, *89*, 032804. [[CrossRef](#)]
32. Elgendy, I.A.; Zhang, W.Z.; He, H.; Gupta, B.B.; Abd El-Latif, A.A. Joint computation offloading and task caching for multi-user and multi-task MEC systems: Reinforcement learning-based algorithms. *Wirel. Netw.* **2021**, *27*, 2023–2038. [[CrossRef](#)]
33. Barabási, A.L.; Bonabeau, E. Scale-free networks. *Sci. Am.* **2003**, *288*, 60–69. [[CrossRef](#)] [[PubMed](#)]
34. Villanueva, R.A.M.; Chen, Z.J. ggplot2-elegant graphics for data analysis. *J. Stat. Softw.* **2017**, *77*, 1–3.
35. Sievert, C. *Interactive Web-Based Data Visualization with R, Plotly, and Shiny*; CRC Press: Boca Raton, FL, USA, 2020.
36. R Core Team. R: A Language and Environment for Statistical Computing Authors R Core Team. Available online: <https://www.r-project.org/> (accessed on 30 November 2021).