Optimized Provisioning of Edge Computing Resources with Heterogeneous Workload in IoT Networks

Article *in* IEEE Transactions on Network and Service Management · January 2019

DOI: 10.1109/TNSM.2019.2894955

5 authors, including:

Hyame Alameddine
Concordia University Montreal
9 PUBLICATIONS **40** CITATIONS
SEE PROFILE

Sanaa Sharafeddine
Lebanese American University
70 PUBLICATIONS **281** CITATIONS
SEE PROFILE

Chadi Assi
Concordia University Montreal
349 PUBLICATIONS **5,147** CITATIONS
SEE PROFILE

Ali Ghrayeb
Texas A&M University at Qatar
238 PUBLICATIONS **4,953** CITATIONS
SEE PROFILE

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2019.2894955, IEEE Transactions on Network and Service Management

Optimized Provisioning of Edge Computing Resources with Heterogeneous Workload in IoT Networks

Nouha Kherraf, Hyame Assem Alameddine, Sanaa Sharafeddine, Chadi Assi, and Ali Ghrayeb

***Abstract*—The proliferation of smart connected Internet of Things (IoT) devices is bringing tremendous challenges in meeting the performance requirement of their supported real-time applications due to their limited resources in terms of computing, storage, and battery life. In addition, the considerable amount of data they generate brings extra burden to the existing wireless network infrastructure. By enabling distributed computing and storage capabilities at the edge of the network, Multi-access Edge Computing (MEC) serves delay sensitive, computationally intensive applications. Managing the heterogeneity of the workload generated by IoT devices, especially in terms of computing and delay requirements, while being cognizant of the cost to network operators requires an efficient dimensioning of the MEC-enabled network infrastructure. Hence, in this paper, we study and formulate the problem of MEC Resource Provisioning and Workload Assignment for IoT services (RPWA) as a Mixed Integer Program (MIP) to jointly decide on the number and the location of edge servers and applications to deploy, in addition to the workload assignment. Given its complexity, we propose a decomposition approach to solve it which consists of decomposing RPWA into the Delay Aware Load Assignment (DALA) sub-problem and the Mobile Edge Servers Dimensioning (MESD) sub-problem. We analyze the effectiveness of the proposed algorithm through extensive simulations and highlight valuable performance trends and trade-offs as a function of various system parameters.**

***Index Terms*—Multi-access edge computing, Internet of Things, 5G, task offloading, Resource allocation, Optimization, Operation research.**

I. Introduction

The number of Internet-connected devices (e.g., smart phones, tablets, smart cameras, industrial sensors, connected cars, smart traffic lights, etc.) is expected to exceed 50 billions by 2020 [1], hence inadvertently realizing the paradigm of the so-called Internet of Things/Everything (IoT/E). Given the immense proliferation of IoT devices, continuous advancements and development of data analytics and networking will empower them with enhanced real-time capabilities [2]. For instance, advances in Radio Frequency Identification (RFID) and Near Field Communication (NFC) technologies made real-time monitoring of almost every entity in a supply chain possible, from inventory tracking to after-sales services. Moreover, sensor technologies have enabled real-time monitoring and processing of traffic flows and vehicles' information in an intelligent traffic ecosystem. The integration of IoT devices in the healthcare domain enabled tracking patients and monitoring their vital signs [3]. However, owing to the shear volume of data these devices generate, they fall short in terms of computing capacity and storage [4], hence restricting their capabilities and hindering the performance of the applications they can support. Traditionally, the cloud has been the go-to solution for providing storage and computing resources to process the vast amount of data generated by IoT devices and to execute the necessary analytics [5], [6]. However, offloading the workload generated by these devices to a remote cloud infrastructure for processing is subject to high communication delays and energy consumption which will eventually violate the latency requirements of real-time IoT applications [7], [8].

Mobile Edge Computing (MEC) has emerged as a new paradigm to overcome the aforementioned challenges. By leveraging a distributed range of computing and storage resources deployed in close proximity to User Equipment (UE), MEC provides the IoT devices the opportunity to offload and run their workload on a wide range of IoT applications deployed on edge servers, hence, supplying them with varying Quality of Service (QoS) requirements [9]. Unlike the traditional centralized cloud, edge servers are collocated with 4G/5G cellular Base Stations (BSs) [10] deployed at the edge of the network. Recently, the term MEC has been redefined as Multi-access Edge Computing, extending its applicability to include new connectivity options such as WiFi, Z-wave and fixed access technologies and, hence, enabling the support of a wider variety of devices and use cases [11]. Another paradigm that has been introduced by Cisco is fog computing which also brings the cloud's intelligence and processing capabilities to the edge of the network. Fog, however, offers a multi-layer cloud computing architecture where fog nodes are deployed in different network tiers (i.e., small base stations, vehicles, wifi access point, and user terminals) [12][13]. The terms fog and edge computing have been used interchangeably in the literature [14]. Further, edge computing capabilities can be enabled at business premises and accessed through wifi Access Points (APs) and are typically known as cloudlets [13]. IoT devices access these edge resources by connecting via an existing wireless access network technology, therefore providing faster response times and saving bandwidth by reducing the load on the network core [7], [8].

In order to enable MEC capabilities, current network infrastructure should be dimensioned to support the deployment of edge servers. However, the cost of such deployment is one of the major obstacles facing network operators given the massive number and spatial spread of IoT devices which are expected to be served within tolerable/low delays [7], [15]. In addition,

This work is supported by NSERC Discovery Grant and by Concordia University.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2019.2894955, IEEE Transactions on Network and Service Management

as the offloaded IoT workloads are required to be processed by different types of applications, usually running on Virtual Machines (VMs) hosted on the edge servers, the decision on the number of instances and the computing resources to assign to each of them becomes challenging and has a direct impact on the response time achieved. Finally, as many IoT devices may be requiring the edge servers capabilities at the same time, efficient and dynamic assignment of their workloads to the hosted applications is required. Since collectively addressing these challenges is a difficult task, the authors of [8] assumed already deployed cloudlets (e.g. edge servers) and addressed the problem of IoT applications placement and workload assignment. The problem of joint application placement on fog nodes and data stream routing from IoT devices to them has been considered recently in [7] with both bandwidth and delay performance guarantees. Task offloading to cloudlets interconnected through a metropolitan wide area network has also been studied in [16], where tasks require virtual functions for their processing. In addition, the optimal placement of cloudlets in a metropolitan area network has been addressed in [17] with the objective of balancing the workload among the deployed cloudlets.

Unlike the work in the literature, we envision an environment with a large number of IoT devices requesting a set of delay-sensitive services (e.g., smart cities, connected cars, industrial control, environmental monitoring, etc.) [18], [19] that can be offered by a wide range of applications. We address the *MEC Resource Provisioning and Workload Assignment for IoT services (RPWA)* problem which consists of jointly solving: 1) *The MEC dimensioning sub-problem* which consists of deciding on the number and the placement of edge servers; 2) *The IoT applications placement sub-problem* which aims at determining the number and the placement of different types of applications' instances to deploy on the edge servers, in addition to deciding on the computing resources (i.e., CPU shares) to allocate to each of them; 3) *The workload assignment sub-problem* which proposes the assignment of the workload generated by IoT devices to the required type of application instance hosted on a suitable edge server and able to achieve its required response time. We formulate the RPWA problem as a Mixed Integer Program (RPWA-MIP) with the objective of minimizing the edge servers deployment cost. As we prove its NP-Hardness, we exploit the inter-dependency existing between the three aforementioned sub-problems composing it, and propose a decomposition approach (RPWA-D) to address its different aspects in a more efficient and scalable strategy. Hence, we divide the RPWA problem into two sub-problems: 1) *The Delay Aware Load Assignment (DALA) sub-problem* which solved the workload assignment sub-problem while deciding on the number and the computing resources to assign to the applications to deploy; and 2) *The Mobile Edge Servers Dimensioning (MESD) sub-problem* which solves the MEC dimensioning sub-problem while deciding on the placement of the applications that needs to be deployed (i.e., provided by DALA) on the provisioned edge servers. Through extensive numerical evaluation, we explore and analyze the different trade-offs existing between the edge servers deployment cost and the workloads of variable QoS requirements which can be served. Under varying parameters, we show that our proposed decomposition approach is efficient, scalable and provide comparable results to those provided by the RPWA-MIP.

The remainder of the paper is organized as follows. Section II presents the literature review. Section III introduces the system model. Section IV defines and formulates the RPWA problem. Section V presents and explains the proposed decomposition approach. Our numerical evaluation is depicted in Section VI. We conclude in Section VII.

II. Literature review

The new concept of MEC has stimulated much research work in the past few years, of which we are going to survey a prime selection of the most closely related.

A. Task offloading and resource allocation in MEC

The authors of [20] have recently presented a survey on exploiting MEC technologies for the realization of IoT applications. The authors in [21] considered the cloudlets placement problem in a Software Defined Networking (SDN)-based IoT network with a focus on minimizing the average cloudlet access delay. They assumed that the SDN control plane manages the routing of IoT devices' requests by commanding a set of SDN-enabled APs. They proposed an enumeration-based algorithm that finds the optimal placement of the cloudlets by evaluating all possible combinations. To reduce the complexity, they devised another ranking-based near-optimal algorithm for the cloudlets placement. The authors, however, did not consider the cloudlets deployment cost.

The work in [22] accounted for the static and dynamic design of an edge cloud network, while respectively considering the absence and the presence of user mobility. Thus, they presented a column generation approach to determine the sites on which the cloudlets have to be installed. Then, they determined the BSs to cloudlets assignment. Finally, the authors addressed the resource allocation problem in terms of determining the placement of each VM required by an end device with respect to its mobility conditions and latency requirement. Although the authors in [22] took into account the cloudlets deployment cost, they did not consider the sharing of VMs between multiple end devices.

The authors in [23] developed an Integer Linear Program (ILP) model for the placement of IoT application services. Unlike [22], the authors considered that VMs could be shared by multiple IoT devices. They solved the model iteratively by considering a new objective at each iteration. These objectives not only addressed the cost efficiency of operating the network (i.e., minimizing the number of active computation nodes and gateways, maximizing the number of admitted applications requests, etc.) but also accounted for the latency reduction (minimizing the hop count). The work in [24] considered the data placement of IoT applications in a fog infrastructure with the objective to minimize the network latency. The authors developed a divide and conquer heuristic in which the fog nodes were weighted and partitioned, resulting in a data placement sub-problem for each partition. Each sub-problem

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2019.2894955, IEEE Transactions on Network and Service Management

was then solved using algorithms implemented in the simulator iFogSim. However, in both works [23] [24], specific latency requirement for each IoT application was overlooked.

Unlike [23], [24], the authors in [7] jointly considered the IoT applications latency and bandwidth requirements. They tackled the applications' requests assignment problem in a fog infrastructure and developed a provisioning model that is responsible for applications' data routing and assignment to the fog nodes. However, they considered that each application has specific hardware requirements, preventing it from being able to be served by any of the fog nodes as some of them may be deprived from the requested hardware resources. The authors in [8] tackled both the resource allocation and IoT application requests assignment problems in an edge infrastructure. They accounted for the limited computing resources of the cloudlets in addition to both network and computation delays of the application requests. They assumed that any IoT application has a specific latency requirement and, unlike [7], can be hosted on top of a VM deployed on any cloudlet. Within this framework, the authors were aiming at minimizing the response time of the applications' requests. Hence, they developed an algorithm that sequentially solves the assignment and resource allocation problems.

The problem of optimal placement of cloudlets in a metropolitan area network, where cloudlets are assumed to be collocated with APs, has been addressed in [17]. Given the complexity of the problem, the authors presented two heuristic solutions. They also considered the users to cloudlet assignment problem to minimize the response time, and noted that routing traffic normally to the closest cloudlet may not always yield a satisfactory solution in terms of response times and, thus, it is necessary to balance the workload among the deployed cloudlets. Task offloading to cloudlets hosting virtual network functions has been considered in [16], and more recently, in [25] for MEC in software defined ultra dense networks.

The resource allocation problem was also considered in the central cloud by the authors of [26], [27]. In [26], the authors considered a cloud resource provisioning scheme to reduce the price the customers have to pay for leasing cloud resources using stochastic optimization. However, in [27], the authors presented a bidding strategy to decide on the customers to serve with the objective of maximizing the cloud provider profit while minimizing the amount of Service Level Agreement Violation (SLA). While in both works the cloud infrastructure was considered as available, we aim in this work at planning and dimensioning the edge cloud infrastructure.

B. Novelty of our work

To the best of our knowledge, the work in the literature focused on solving at most two of the aforementioned subproblems; MEC dimensioning, IoT placement and workload assignment. Further, most of the work mentioned above assumed a homogeneous workload or a single user equipment. While the work in [8] considered the heterogeneity of applications' requests in an IoT environment, they, however, did not address the MEC dimensioning problem and assumed a network where the edge servers are already placed. The novelty of our work is, therefore, manifested by jointly solving the three above mentioned problems and considering a large scale IoT environment with heterogeneous applications' requests.

III. System Model

A. MEC-enabled Smart Environment

The system model we target resembles the scenario depicted in Fig. 1. A metropolitan area encompassing a massive number of Internet-enabled devices generating requests to diverse IoT applications to support services such as smart health, intelligent transportation, and environmental monitoring. We assume a multi-access edge cloud where IoT applications are hosted on edge servers, fog nodes or cloudlets, that are accessible through a set of WiFi APs (or cellular BSs). We also assume a backbone network infrastructure available to interconnect the APs to each other. Hence, some APs may not need to host edge servers or may host an edge server but not support a particular IoT application. This enables a graceful, rather than ubiquitous, cost-effective deployment of edge servers. We further assume that a wide range of devices spatially distributed throughout the smart environment generate workload by requesting given IoT services provided by corresponding application types hosted on selected edge servers. This can lead to scenarios where a given device can access applications on edge servers co-located with APs other than its serving AP. Therefore, the various edge servers hosting IoT applications should be accessible by devices from various locations via the backbone network infrastructure. Formally, the network is abstracted as a graph $G(N, E)$, where N is the set of nodes; $N = L \cup R$ is composed of a set of APs (or BSs) dispersed at various accessible locations ($l \in L$) in the network, and R represents the set of backbone network equipment (e.g., routers, switches, etc.). All nodes in the network are interconnected with a set E of communication links. Each IoT device connects to the closest AP, and can utilize services provided by IoT applications deployed at either an edge server attached to its serving AP or an edge server attached to another AP that is accessible through the network backbone. Let M be the set of edge servers, each with a computing capacity c_m. An edge server m once deployed at AP l will incur a location-dependent deployment cost π_m^l. Let A be the set of all IoT applications; an application $a \in A$ will be hosted on a VM running on one of the edge servers. Each application provides certain functions for a particular IoT service and, hence, is subject to workload generated from IoT devices that are subscribed to this particular service; for example, a smart camera may request its video streams to be processed by an application providing rendering or surveillance functionality. Applications are classified into different types T (i.e., video processing, face recognition, etc.) and, thus, we define $\mu_a^t \in \{0, 1\}$ to denote that an application a is of type t. In addition, each application requires minimum computing resources p_{min}^a to run and efficiently handle the computation of the minimum load of the IoT devices accessing it [28]. In order to avoid assigning all the computing resources of an edge server to one IoT application, we denote by p_{max}^a

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2019.2894955, IEEE Transactions on Network and Service Management

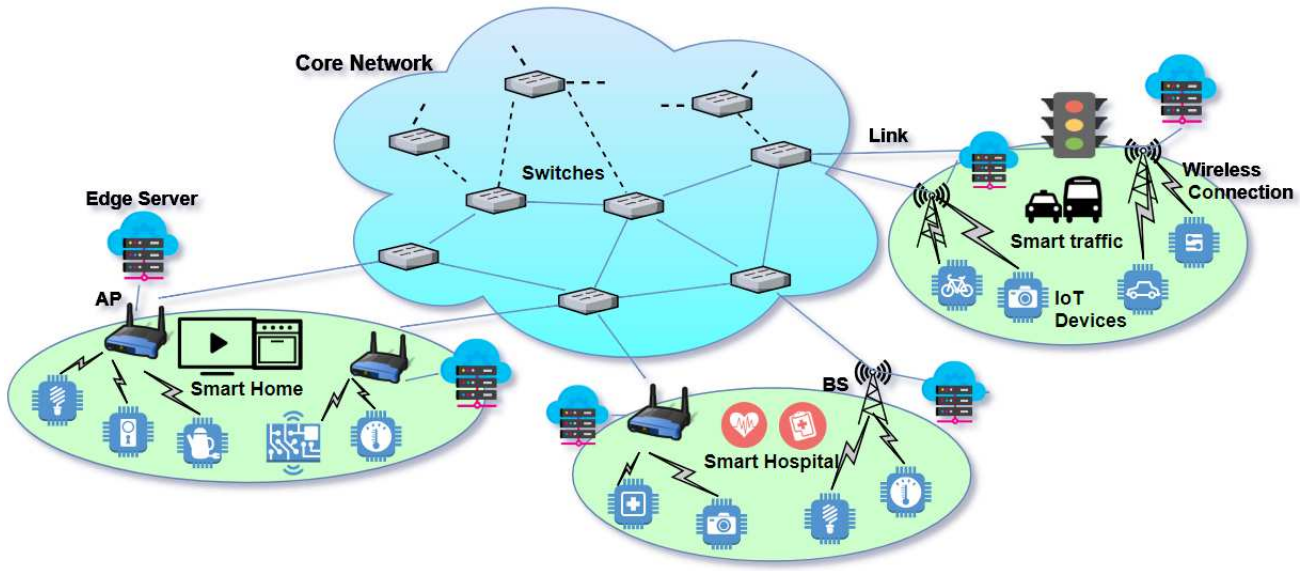


Fig. 1: MEC-enabled smart environment.

the maximum computing resources that can be assigned to an IoT application $a \in A$. More precisely, p^a_{min} and p^a_{max} are the minimum and maximum computing resources that can be assigned to the VM hosting the IoT application a. Further, given that some services require ultra-low latency (e.g., tactile Internet or industrial control), each application of type $t \in T$ is subject to a maximum allowable response time δ_t to ensure satisfactory QoS. The aggregate workload requesting the service provided by applications of type t generated from the devices located within the coverage area of AP deployed at location l is assumed to follow a Poisson process, with an arrival rate λ^t_l (requests/sec), where each request has a given average computing size w_t (e.g., CPU cycles needed to complete the task). This workload can be processed by any hosted application of the requested type t. Further, a given application may receive workload requests from different locations in the network. The higher the intensity of the offloaded workload on a particular application, the more computing resources this application should be provided in order to keep the processing and queuing delays of the corresponding tasks low. Thus, we denote by p_a (measured in cycles/second), the computing capacity assigned to the IoT application a through the VM hosting it. Further, given that an application hosted on a VM running on an edge server could be shared by multiple loads coming from different locations $l \in L$ following a Poisson process, and the load computation time is exponentially distributed, each application is modeled as an M/M/1 queue [29], with an aggregate arrival rate of IoT requests and a service rate based on the processing capacity p_a of each application. Finally, as mentioned earlier, a request can be either processed by its home edge server (the edge server that is co-located with its serving AP l) or processed by a different edge server at AP deployed at location l'. Therefore, we denote by $h^{l'}_l$ the network delay representing the delay incurred by routing the workload request from the serving AP at location l to the assigned edge server at location l'.

B. Problem Description

Providing efficient IoT devices workload offloading and processing requires upgrading existing networks to become MEC-enabled. Thus, based on the proposed system model (Section III-A) and while accounting for stringent latency requirement of the offloaded workload, we study and explore three interleaving challenges intrinsic for a successful cost-efficient dimensioning of current networks to become MEC-enabled. Hence, we exploit the following sub-problems:

1) *The MEC dimensioning sub-problem* which consists of deciding on the number and the placement of edge servers. Such decision is affected by the workload to be offloaded from the different existing locations and their required response times. Further, the computing capacities of the edge servers to deploy at different locations may also be variable and subject to the workload which it will be processing. Hence, as edge servers of variable computing capacities may be available for deployment, the MEC dimensioning sub-problem seeks at choosing those with enough computing capacity to deploy at specific locations. Such decision has a direct impact on the deployment cost which increases with the edge servers computing resources.

2) *The IoT applications placement sub-problem* which aims at determining the number and the placement of different types of applications' instances to deploy on the edge servers. Such decision is coupled to the size of the workload requesting a specific application type from each location and its latency requirement. For instance, as each application instance can only be deployed on exactly one edge server, the choice of the edge server that will host it will affect the response time of the workload that it will be processing. Such response time includes the network delay (e.g., if the application was deployed on an edge server collocated to an AP/BS other than its serving AP/BS), in addition to the processing and queuing delays. These latter are respectively dependent on

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2019.2894955, IEEE Transactions on Network and Service Management

the processing capacity assigned to the application and the amount of workloads assigned to this specific application. Thus, the IoT applications placement sub-problem is also required to decide on the processing capacity p_a to assign to the application instance to deploy. Note that, the choice of p_a limits the choice of edge servers that can host the application instance, as some of them may not have enough computing resources left to host it. In fact, the computing capacity of edge servers should be optimally allocated to different types of applications in order to successfully meet the delay requirements of the assigned workloads. Further, the processing resources (p_a) assigned to the application instances to deploy impacts the deployment cost as more edge servers may be needed to accommodate those applications.

3) *The workload assignment sub-problem* which consists of assigning the workload generated by IoT devices to the required type of application instance hosted on a suitable edge server, able to meet its response time. It is important to note that assigning the workloads to the closest edge server may not yield an optimal or feasible solution. In fact, the closest edge server collocated to the serving AP/BS may not be hosting the required type of application. Alternatively, the processing capacity assigned to the required application hosted on the edge server may not be enough to serve the workload with respect to its latency requirement. Hence, as we note the inter-dependencies between the three aforementioned sub-problems, we define a joint problem entitled *MEC Resource Provisioning and Workload Assignment for IoT services (RPWA)* which we mathematically formulate next.

IV. RPWA - A MIXED INTEGER PROGRAM (RPWA-MIP)

A. Problem Definition

Definition 1. *Given $G(N, E)$, a set M of edge servers, a set A of IoT applications of different types, and a set of IoT devices requesting their workloads to be offloaded and processed by a specific application type within a determined response time δ_t, determine the lowest cost deployment of edge servers and IoT applications in $G(N, E)$, the processing capacity to allocate to the deployed applications, in addition to the assignment of workloads to these latter with respect to their latency requirements.*

B. Problem Formulation

Table I delineates the parameters used throughout the formulation of RPWA-MIP presented below. We define a variable x_m^l to determine whether edge server $m \in M$ is deployed at location $l \in L$.

$$x_m^l = \begin{cases} 1 \text{ if edge server } m \in M \text{ is deployed at location } l \in L, \\ 0 \text{ otherwise.} \end{cases}$$

Our objective is to minimize the edge servers deployment cost while meeting the delay requirements of the workloads requesting to be processed by a specific type of IoT applications:

$$Minimize \quad \sum_{l \in L} \sum_{m \in M} x_m^l (\pi_l + k c_m) \tag{1}$$

Network Inputs	
$G(N, E)$	Network of N nodes where $N = L \cup R$ and E links connecting them.
L	Set of locations mounted with APs/BSs.
R	Set of backbone network equipment.
M	Set of edge servers to be deployed in $G(N, E)$.
A	Set of IoT applications to be hosted in $m \in M$.
T	Set of applications' types.
$c_m \in \mathbb{R}^+$	Processing capacity of edge server $m \in M$.
$\pi_l \in \mathbb{Z}^+$	Setup cost of an edge server at location $l \in L$.
$k \in \mathbb{Z}^+$	Cost of a unit of processing capacity.
$\mu_a^t \in \{0, 1\}$	Parameter which depicts that application $a \in A$ is of type $t \in T$ (1) or not (0).
$\delta_t \in \mathbb{R}^+$	Maximum allowable response time when utilizing an application of type $t \in T$.
$p_{min}^a \in \mathbb{R}^+$	Minimum processing capacity required by application $a \in A$.
$p_{max}^a \in \mathbb{R}^+$	Maximum processing capacity that can be assigned to application $a \in A$.
$\lambda_l^t \in \mathbb{Z}^+$	Arrival rate of requests for an application of type $t \in T$ generated by IoT devices associated to AP/BS at location $l \in L$.
$w_t \in \mathbb{Z}^+$	Average number of CPU cycles per request for an application of type t .
$h_l^{l'} \in \mathbb{R}^+$	Network delay of a request from its home edge server at $l \in L$ to its assigned edge server at $l' \in L$.

TABLE I: Parameters of RPWA-MIP.

In order to meet our objective, several constraints, that we elucidate in the following, have to be respected. Let y_a^{lm} be a decision variable that determines whether application $a \in A$ is placed on edge server $m \in M$ deployed at location $l \in L$.

$$y_a^{lm} = \begin{cases} 1 \text{ if IoT application } a \text{ is placed on } m \text{ at location } l, \\ 0 \text{ otherwise.} \end{cases}$$

In order to simplify some of the constraints, we define σ_a^l to determine whether application a is placed at location l.

$$\sigma_a^l = \begin{cases} 1 \text{ if IoT application } a \text{ is placed at location } l, \\ 0 \text{ otherwise.} \end{cases}$$

z_{lt}^a is another decision variable that indicates whether the workload generated by IoT devices at location $l \in L$ demanding an application of type $t \in T$ are mapped to application $a \in A$.

$$z_{lt}^a = \begin{cases} 1 \text{ if workload generated by devices at location } l \text{ and} \\ \text{demanding an application of type } t \text{ is mapped to } a, \\ 0 \text{ otherwise.} \end{cases}$$

Further, let $p_a \in \mathbb{R}^+$ specifies the amount of computing resources assigned to application $a \in A$. Hence, the RPWA problem constraints can be classified as follows:

Placement of edge servers

First, we need to guarantee that each edge server $m \in M$ is deployed on at most one location $l \in L$ (Eq.(2)), that is: $\sum_{l \in L} x_m^l = 1$ if edge server m is deployed at location l, and 0 otherwise. Further, Eq.(3) guarantees that each location

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2019.2894955, IEEE Transactions on Network and Service Management

can host at most one edge server; hence, $\sum_{m\in M} x_m^l = 1$ if location l is hosting one edge server, and 0 otherwise.

$$\sum_{l\in L} x_m^l \leq 1 \quad \forall m \in M \tag{2}$$

$$\sum_{m\in M} x_m^l \leq 1 \quad \forall l \in L \tag{3}$$

Placement of applications on edge servers

Once the edge servers are deployed, we provision IoT applications and assign computing resources to them. Eq.(4) ensures that each application $a \in A$ is placed on at most one edge server $m \in M$. Thus, if application a is placed on an edge server m, then $\sum_{l\in L}\sum_{m\in M} y_a^{lm} = 1$, and 0 otherwise. Further, each application's computing resource is guaranteed a minimum value p_{min}^a (Eq.(5)), and limited to a maximum value p_{max}^a (Eq.(6)).

$$\sum_{l\in L}\sum_{m\in M} y_a^{lm} \leq 1 \quad \forall a \in A \tag{4}$$

$$p_a \geq \sum_{m\in M}\sum_{l\in L} y_a^{lm} p_{min}^a \quad \forall a \in A \tag{5}$$

$$p_a \leq \sum_{m\in M}\sum_{l\in L} y_a^{lm} p_{max}^a \quad \forall a \in A \tag{6}$$

Constraint (7) guarantees that all IoT applications provisioned on an edge server $m \in M$ do not exceed the capacity c_m of m. Constraint (8) guarantees that each application $a \in A$ can only be placed on a deployed edge server $m \in M$.

$$\sum_{l\in L}\sum_{a\in A} p_a y_a^{lm} \leq c_m \quad \forall m \in M \tag{7}$$

$$y_a^{lm} \leq x_m^l \quad \begin{matrix}\forall l\in L\\ \forall m\in M\\ \forall a\in A\end{matrix} \tag{8}$$

Further, (9) ensures that if $y_a^{lm} = 1$ then $\sigma_a^l = 1$.

$$\sigma_a^l = \sum_m y_a^{lm} \quad \begin{matrix}\forall l\in L\\ \forall a\in A\end{matrix} \tag{9}$$

Workload assignment

Once applications are placed, the workload generated by devices associated to an AP/BS at location $l \in L$ is assigned to an IoT application of the requested type $t \in T$. Thus, constraint (10) ensures that z_{lt}^a gets a value, for some a, when $\lambda_l^t > 0$; H is a big integer number and $\lambda_l^t << H$.

$$\sum_{a\in A} z_{lt}^a \geq \frac{\lambda_l^t}{H} \quad \begin{matrix}\forall l\in L\\ \forall t\in T\end{matrix} \tag{10}$$

Constraint (11) guarantees that the generated load from location l requesting application of type t is mapped to at most one IoT application a:

$$\sum_{a\in A} z_{lt}^a \leq 1 \quad \begin{matrix}\forall l\in L\\ \forall t\in T\end{matrix} \tag{11}$$

In addition, we need to make sure that each load is mapped to an application $a \in A$ providing the same requested type t:

$$z_{lt}^a \leq \mu_a^t \lambda_l^t \quad \begin{matrix}\forall t\in T\\ \forall l\in L\\ \forall a\in A\end{matrix} \tag{12}$$

Note that Eq.(12) guarantees that $z_{lt}^a = 0$ if there are no requests for an application of type t coming from location l, i.e., $\lambda_l^t = 0$. Constraint (13) ensures that requests are only mapped to applications that are deployed on an edge server $m \in M$ placed at some location $l \in L$ and constraint (14) prevents hosting an application a on an edge server m if it is not processing any load.

$$z_{lt}^a \leq \sum_{m\in M}\sum_{l'\in L} y_a^{l'm} \quad \begin{matrix}\forall l\in L\\ \forall t\in T\\ \forall a\in A\end{matrix} \tag{13}$$

$$\sum_{m\in M}\sum_{l\in L} y_a^{lm} \leq \sum_{l\in L}\sum_{t\in T} z_{lt}^a \quad \forall a \in A \tag{14}$$

Delay Constraints

Offloaded IoT traffic experiences delays consisting of access delays from IoT device to the serving AP (or BS), network delays if traffic is to be routed from the serving AP to the AP where the edge server is hosted, and finally server delays incurred at the edge server of the receiving application and that constitutes queuing and processing delays. We assume negligible access delays and focus on the network delays and server delays. To guarantee delay requirements for each IoT service provided by applications of type t, we constrain in Eq.(15) the total delay experienced by the traffic to a given target response time δ_t.

$$2d_n^{l,t} + d_s^{l,t} \leq \delta_t \quad \begin{matrix}\forall l\in L\\ \forall t\in T\end{matrix} \tag{15}$$

where $d_n^{l,t}$ represents the network delays experienced by a workload generated from location l requesting an application of type t and assigned to an IoT application hosted in a remote location, and is given by:

$$d_n^{l,t} = \sum_{a\in A}\sum_{l'\neq l} h_l^{l'} z_{lt}^a \sigma_a^{l'} \quad \begin{matrix}\forall l\in L\\ \forall t\in T\end{matrix} \tag{16}$$

$d_s^{l,t}$ depicts the server delay that constitutes the queuing and processing delays that the workload generated from location l experiences at the edge server of the receiving application of type t. As mentioned earlier, given the workload assigned from various locations l' to each application, we model an application a as an M/M/1 queuing system with aggregate request arrival rate of $\sum_{l'\in L} z_{l't}^a \lambda_{l'}^t$ and service rate of $\frac{p_a}{w_t}$, where p_a is the processing capacity in cycles per second assigned to application a and w_t is the average request size in cycles. Therefore,

$$d_s^{l,t} = \sum_{a\in A} z_{lt}^a \left(\frac{1}{\frac{p_a}{w_t} - \sum_{l'\in L} z_{l't}^a \lambda_{l'}^t}\right) \quad \begin{matrix}\forall l\in L\\ \forall t\in T\end{matrix} \tag{17}$$

Eq. (15) after some manipulation can be rewritten as:

$$\sum_{a\in A} z_{lt}^a \frac{p_a}{w_t} - \sum_{a\in A}\sum_{l'\in L} z_{lt}^a z_{l't}^a \lambda_{l'}^t \geq \frac{1}{\delta_t - 2\sum_{a\in A}\sum_{l'\neq l} h_l^{l'} z_{lt}^a \sigma_a^{l'}} \quad \begin{matrix}\forall l\in L\\ \forall t\in T\end{matrix} \tag{18}$$

To maintain a stable queue at the application, we force the service rate to be greater than the arrival rate as per below:

$$\frac{p_a}{w_t} - \sum_{l'\in L} z_{l't}^a \lambda_{l'}^t \geq 0 \quad \begin{matrix}\forall a\in A\\ \forall t\in T\end{matrix} \tag{19}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2019.2894955, IEEE Transactions on Network and Service Management

The above model is a mixed integer non-linear program. Appendix A presents linearization of Eq.(7) and Eq.(18).

C. Complexity Analysis

The RPWA-MIP formulation is complex and the model is clearly hard to solve even for a small network (Section VI). In fact, the complexity of the RPWA problem can be highlighted through the complexity of the different sub-problems it solves. For instance, the MEC dimensioning and the IoT applications placement sub-problems combined can be proven as NP-Hard via a reduction from the capacitated facility location problem [30] (known to be NP-Hard) where the facilities are the edge servers to be deployed at locations $l \in L$ and the customers are the applications to place on the deployed edge servers. Further, the NP-Hard generalized assignment problem [31] can be reduced to the workload assignment sub-problem where the workloads constitute the items that need to be assigned to bins representing the IoT applications. Hence, the workload assignment sub-problem is also NP-Hard. Thus, the RPWA problem is NP-Hard as it combines three NP-Hard sub-problems. Given its complexity, we present in the following an efficient decomposition approach to solve it.

V. RPWA-D: A Decomposition Approach

As addressing the MEC dimensioning, the IoT applications placement and the workload assignment sub-problems jointly is challenging, we investigate the inter-dependency tightening these three sub-problems together in the aim of exploring a more efficient approach to solve them. Thus, we first notice that the workload assignment sub-problem highly couples the MEC dimensioning sub-problem and the IoT applications placement sub-problem which makes it difficult to address each of these three sub-problems independently. By investigating the workload assignment sub-problem, we observe that the placement of the applications on edge servers has a direct impact on the network delay experienced by the workload assigned to the application. Further, the decision on the computing resources to allocate to each application is dependent on the size of the workload it will be processing and affects the server delay of this latter. In fact, the response time (δ_t) of the IoT service requested by the offloaded workloads is to be met as per Eq.(15). Hence, we re-evaluate the response time constraint in Eq.(15) given that it is a critical constraint linking the three aforementioned sub-problems. Eq.(15) is rewritten as follows:

$$d_s^{l,t} \leq \delta_t - 2d_n^{l,t} \quad \begin{matrix} \forall l \in L \\ \forall t \in T \end{matrix} \tag{20}$$

By substituting $d_s^{l,t}$ given in Eq.(17) into Eq.(20), we obtain:

$$\sum_{a \in A} z_{lt}^a \left(\frac{1}{\frac{p_a}{w_t} - \sum_{l' \in L} z_{l't}^a \lambda_{l'}^t} \right) \leq \delta_t - 2d_n^{l,t} \quad \begin{matrix} \forall l \in L \\ \forall t \in T \end{matrix} \tag{21}$$

From (21), we notice that if $d_n^{l,t}$ is known, it will be easy to decouple the workload assignment sub-problem from the IoT applications placement and MEC dimensioning sub-problems as the placement of the Iot applications and edge servers will no longer affect the response time requirement (δ_t). To resolve this, we assume a maximum network delay experienced by the load ($d_n^{max} = \max\ (h_l^{l'}),\ \forall l,l' \in L$), to guarantee that any obtained solution will meet the response time requirement. Although this assumption simplifies the decomposition of our problem, it may result in over-provisioning the edge servers as it assumes that some loads incur higher network delays from what they actually experience. Eq.(21) can then be rewritten as follows:

$$\sum_{a \in A} z_{lt}^a \left(\frac{1}{\frac{p_a}{w_t} - \sum_{l' \in L} z_{l't}^a \lambda_{l'}^t} \right) \leq \delta_t' \quad \begin{matrix} \forall l \in L \\ \forall t \in T \end{matrix} \tag{22}$$

where $\delta_t' = \delta_t - 2d_n^{max}$ and represents the maximum server delay allowed to meet the response time requirement. Eq.(22) shows that given the maximum network delay, one can decide on the computing resources p_a to assign to each application a as well as the workload mapped to this application. Determining the number of needed applications of the same type and the computing resources (p_a) to assign to each of those applications based on a *maximum* server delay (δ_t'), which may be experienced by all workloads requesting the same application type, simplifies the IoT applications placement sub-problem to simply respecting the edge servers capacities. In fact, the applications and the edge servers can then be placed at any location without incurring any violation to the response time requirement. Hence, since applications placement on edge servers is dependent on the computing capacities of these edge servers and their number, consequently, the total deployment cost is affected. Thus, the MEC dimensioning sub-problem can take care of the IoT applications placement, if fed with those to be deployed under the objective of minimizing the deployment cost. The workload assignment sub-problem can, hence, take care of deciding on the number and the computing resources to be assigned to the applications to deploy. Thus, the decisions initially taken by the IoT applications placement sub-problem are divided between the two other sub-problems (i.e., MEC dimensioning and workload assignment sub-problems). Therefore, in the following, we decouple the RPWA problem and decompose it into two different sub-problems:

1) The Delay Aware Load Assignment (DALA) sub-problem that decides on the workload assignment to the IoT applications while determining the number of needed applications and their computing resources with respect to the response time requirement.

2) The Mobile Edge Servers Dimensioning (MESD) sub-problem which determines the placement of the applications and edge servers.

A. The Delay Aware Load Assignment (DALA)

Definition 2. *Let $G(N, E)$ denotes the network, R_t represents the set of all workloads demanding the service of an IoT application of type t (each $r \in R_t$ denotes a load with rate λ_r requests/sec for application of type t). Let A_t depicts the set of all IoT applications of type t, maximize the fraction of workloads that can be admitted to the network while determining the number of applications to be deployed in $G(N, E)$ and*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2019.2894955, IEEE Transactions on Network and Service Management

the assignment of the admitted workloads to those applications with respect to the response time requirement.

Network Inputs	
A_t	Set of applications of type t.
R_t	Set of all workloads demanding the service of an application of type t.
$\lambda_r \in \mathbb{Z}^+$	Arrival rate of requests of workload $r \in R_t$.
$w_t \in \mathbb{Z}^+$	Average number of computing cycles demanded by one request $r \in R_t$.
$\delta'_t \in \mathbb{R}^+$	Maximum tolerated server delay by an application of type t.

TABLE II: Parameters of the DALA-MIP.

We delineate in Table II the parameters used throughout our formulation. The remaining ones are as defined in Table I. As the main objective of the RPWA problem is to minimize the deployment cost of edge servers, a possible objective for the DALA sub-problem is to minimize the sum of computing resources assigned to the applications to deploy, as it will reduce the number of edge servers needed to host those applications and hence, minimize the deployment cost. While this is a possible objective and comes aligned with the definition of the RPWA problem, it requires admitting all the workloads; this might lead to infeasibility for the same instances which are can be solved by the RPWA-MIP. This is because DALA considers the maximum network delay (d_n^{max}) when searching for a solution, which tightens up the server delay (Eq.22) and hence, the solution space. Thus, to obtain a solution for the same instances of RPWA, we define DALA under the objective of maximizing the fraction of the admitted workloads. Hence, we define $\alpha_r \in [0-1]$ to depict the fraction of the load that can be admitted to the network and we depict the objective of DALA in Eq.(23).

$$\text{Maximize} \quad \sum_{r \in R_t} \lambda_r \alpha_r \tag{23}$$

This objective is subject to several constraints, as elaborated in the sequel. We introduce the decision variable $\rho^a \in \{0,1\}$ to depict whether an application $a \in A_t$ is assigned at least one workload to process.

$$\rho^a = \begin{cases} 1 \text{ if application } a \text{ is used,} \\ 0 \text{ otherwise.} \end{cases}$$

We define a new variable $z_r^a \in \{0,1\}$ to specify whether a workload $r \in R_t$ is mapped to application $a \in A_t$ as follows:

$$z_r^a = \begin{cases} 1 \text{ if workload } r \text{ is mapped to application } a, \\ 0 \text{ otherwise.} \end{cases}$$

In addition, we define $p_a \in \mathbb{R}^+$ to depict the computing resources to be allocated to application $a \in A_t$. An application is used and should be deployed in the network if at least one workload is mapped to it as specified in Eq.(24).

$$\rho^a \geq z_r^a \quad \begin{matrix} \forall a \in A_t \\ \forall r \in R_t \end{matrix} \tag{24}$$

A workload $r \in R_t$ can be mapped to and processed by exactly one application as determined by Eq.(25).

$$\sum_{a \in A_t} z_r^a \leq 1 \quad \forall r \in R_t \tag{25}$$

A workload $r \in R_t$ should be processed by an application $a \in A_t$ that has enough computing resources to process it without violating the server delay requirement as depicted in Eq.(26).

$$d_s^r \leq \delta'_t \quad \forall r \in R_t \tag{26}$$

where d_s^r represents the server delay (processing and queuing delays) and is determined by (27).

$$d_s^r = \sum_{a \in A_t} z_r^a \left(\frac{1}{\frac{p_a}{w} - \sum_{r' \in R_t} z_{r'}^a \lambda_{r'} \alpha_{r'}} \right) \quad \forall r \in R_t \tag{27}$$

Further, the queue of each application should remain stable. That is, the average service rate of the IoT application $a \in A_t$ should be larger than the aggregate average arrival rates of all workloads mapped to application a as given in Eq.(28).

$$\frac{p_a}{w} - \sum_{r' \in R_t} z_{r'}^a \lambda_{r'} \alpha_{r'} > 0 \quad \forall a \in A_t \tag{28}$$

Finally, the computing resources assigned to an application should be at least equal to its minimum required computing capacity as depicted in Eq.(29) and at most equal to $p_m a x^a$ as specified in Eq.(30)

$$p_a \geq p_{min}^a \rho^a \quad \forall a \in A_t \tag{29}$$

$$p_a \leq p_{max}^a \rho^a \quad \forall a \in A_t \tag{30}$$

Eq.(23) and Eq.(26) are non linear and can be easily linearized (Appendix B). We note that DALA is a MIP that can run in multiple threads where each thread finds the solution for one application type.

B. The Mobile Edge Servers Dimensioning (MESD)

Definition 3. *Given a network $G(N,E)$, a set A of IoT applications and a set M of edge servers, determine the number and the placement of edge servers in $G(N,E)$, in addition to the placement of the applications on the deployed edge servers such that the total deployment cost is minimized.*

Network Inputs	
$\bar{A}$	Set of applications to deploy.
$p_a \in \mathbb{R}^+$	Computing resources of application $a \in A$.

TABLE III: Parameters of the MESD-IP.

Table III presents the parameters used. The remaining parameters are as defined in Table I. In order to formulate the MESD sub-problem, we define the decision variable $x_m^l \in \{0,1\}$ to specify whether the edge server $m \in M$ is deployed at location $l \in L$ as follows:

$$x_m^l = \begin{cases} 1 \text{ if edge server } m \text{ is deployed at location } l, \\ 0 \text{ otherwise.} \end{cases}$$

In addition, we introduce a decision variable $y_a^m \in \{0,1\}$ to indicate whether application $a \in \bar{A}$ is placed on edge server $m \in M$ as follows:

$$y_a^m = \begin{cases} 1 \text{ if application } \ a \text{ is placed on edge server } m, \\ 0 \text{ otherwise.} \end{cases}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2019.2894955, IEEE Transactions on Network and Service Management

The objective of the MESD sub-problem is to minimize the deployment cost of the edge servers (Eq.(31)).

$$Minimize \quad \sum_{l \in L} \sum_{m \in M} x_m^l (\pi_l + k c_m) \tag{31}$$

Subject to the following constraints. First, an edge server can be placed on at most one location Eq.(32).

$$\sum_{l \in L} x_m^l \leq 1 \quad \forall m \in M \tag{32}$$

Similarly, a location $l \in L$ can host at most one edge server $m \in M$ as depicted in Eq.(33).

$$\sum_{m \in M} x_m^l \leq 1 \quad \forall l \in L \tag{33}$$

In addition, an application can be deployed on exactly one edge server as specified in Eq.(34).

$$\sum_{m \in M} y_a^m = 1 \quad \forall a \in \bar{A} \tag{34}$$

Finally, Eq.(35) guarantees that the edge servers capacities are not violated and that the applications are only hosted on placed edge servers.

$$\sum_{a \in \bar{A}} y_a^m p_a \leq c_m \sum_{l \in L} x_m^l \quad \forall m \in M \tag{35}$$

C. Decomposition Algorithm

We design and implement the RPWA decomposition solution (RPWA-D) to solve the RPWA problem. Our proposed decomposition approach captures the collaboration between the DALA and the MESD sub-problems. Thus, we depict in Fig.2 a flowchart detailing the steps of RPWA-D. In fact, RPWA-D accounts for the independency that exists between the applications types. For instance, the assignment of the workloads requesting the service provided by a specific type of applications and the computing resources to be assigned to those applications have no impact on the same decisions made for any other type of applications. Hence, several instances of the DALA-MIP are executed as parallel threads in order to provide a workloads assignment and application computing resources determination for each type of applications. Thus, RPWA-D pre-processes the simulation data by categorizing it by the type of application it is requesting. It then instantiate a thread to execute the DALA-MIP for each application type. Once the execution of all the DALA-MIP threads is finalized, their solutions are processed to capture the application instances that need to be deployed on edge servers (i.e., the application instances that were set as used by the DALA-MIP threads ($\rho_a = 1$)), in addition to their assigned computing resources (p_a). Those latter are supplied to MESD-IP as parameters. The MESD-IP is then executed in order to decide on the number and the location of edge servers to place, in addition to the placement of the applications which need to be deployed. Thus, with the help of the DALA-MIP, the MESD-IP provides the minimum edge servers deployment cost.

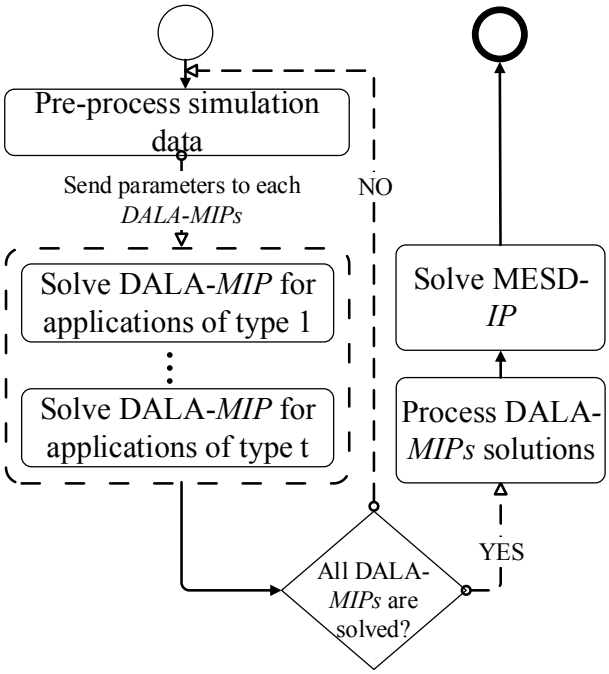


Fig. 2: Flowchart of RPWA-D.

VI. Numerical Evaluation

We conduct an extensive empirical study to evaluate the performance of the RPWA-MIP against our decomposition approach (RPWA-D). We also explore the efficiency of RPWA-D under varying parameters and study the trade-off between the edge servers deployment cost and the percentage of the workloads that can be admitted. Our numerical evaluation is conducted using IBM ILOG CPLEX Optimization Studio version 12.8.

A. Evaluation setup

In our tests, we consider networks of different sizes where we vary the number of locations hosting APs/BSs (L) and the number of edge servers that can be collocated to the existing APs/BSs (M). However, we consider that each edge server $m \in M$ has a computing capacity $c_m = 6GHz$ (unless stated otherwise) and a deployment cost normalized to 8 units. In addition, we perform our tests under different number of available IoT applications (variable A) of 4 different types ($T = 4$) that belong to the same industry vertical and hence, may require the same QoS (i.e., response time). Thus, we delineate in Table IV the different industry verticals accounted for in our tests, and present the range of their required response times in addition to the one used to set the value of δ_t. Note that, we consider that each IoT application $a \in A$ requires a minimum processing capacity of $p_{min}^a = 1.7GHz$ and can consume up to $p_{max}^a = 1.9GHz$ of computing resources. We account for IoT devices scattered at different locations within the considered network. Each of these IoT devices is connected to an AP/BS available at its location and is requesting the service of a determined type of application. We evaluate variable arrival rates (variable λ_l^t) of the IoT devices

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2019.2894955, IEEE Transactions on Network and Service Management

requests per second for different application types spread at different locations of the network. However, we assume an average of $w_t = 2 \times 10^6$ CPU cycles/request for a defined application type $t \in T$.

Industry Vertical	Allowable response times (ms)	Applied response time (δ_t) (ms)
Tactile Internet	1 - 10	5
Factory Automation	0.25 - 10	10
Smart Grids	3 - 20	20
Intelligent transportation Systems (ITS)	10 - 100	50
Tele Surgery	$\leq$ 250	110

TABLE IV: QoS of different industry verticals [32], [33].

B. RPWA-MIP vs. RPWA-D

We first study the performance of RPWA-D and compare it against RPWA-MIP in terms of optimality (e.g., deployment cost) and scalability (e.g., execution time) as we vary the number of locations in the network. In fact, increasing the number of locations depicts an increase in the workloads as more IoT devices requests originating from the added locations are to be accounted for. Hence, we consider a maximum of $M = 10$ edge servers and $A = 4$ applications of $T = 4$ different types belonging to a factory automation vertical ($\delta_t = 10ms$) that can be deployed in the network. In addition, we assume $\lambda_l^t = 60$ requests/sec originating from each location $l \in L$ and requesting the service of an application of type $t \in T$. The maximum network delay is set to $4ms$. Our results are presented in Table V.

Instance $< L, M, T, A >$	RPWA-MIP		RPWA-D		
	Cost (units)	*Execution Time (ms)*	*Cost (units)*	*Execution Time (ms)*	*Load admitted (%)*
L=5, M=10, T=4, A=4	16	69	16	57	100
L=7, M=10, T=4, A=4	16	90	16	60	100
L=15, M=10, T=4, A=4	-	-	16	90	50
L=15, M=10, T=4, A=12	-	2days+	32	661	100

TABLE V: Evaluation of RPWA-MIP vs. RPWA-D.

1) Optimality Gap: Table V depicts that for a small number of locations ($L = 5$ and $L = 7$), RPWA-D is able to provide the optimal cost of 16 units supplied by RPWA-MIP while admitting all the workloads. Note that, this cost remained constant for $L = 5$ and $L = 7$ given that the deployed edge servers when $L = 5$ had enough free computing resources to accommodate the additional applications needed to handle the extra workloads generated from the two added locations (when $L = 7$). In other words, no extra edge servers were needed to be deployed. As we increase the number of locations to $L = 15$, RPWA-MIP fails to give a feasible solution as it is constrained by admitting all the generated workloads and the fixed number of applications ($A = 4$) that can be deployed. However, RPWA-D admitted 50% of the workload with a cost of 16 units. As $A = 4$ applications were not sufficient for RPWA-MIP to provide a solution, we performed a final test where we increased the number of applications from $A = 4$ to $A = 12$ for the same number of locations (e.g., $L = 15$). However, this resulted in a large increase of the size of the problem and RPWA-MIP failed to give a solution even after 2 days of execution. However, RPWA-D was able to admit all the loads.

2) Execution Time: In order to evaluate the scalability of the proposed methods, we delineates in Table V the execution time of RPWA-MIP and RPWA-D. As the size of the problem increases with the increase of the number of locations and the number of applications, finding a solution for the problem becomes more challenging, and hence, the runtime of both methods increases exponentially. However, Table V clearly shows that the increase of the runtime of RPWA-D is at a slower pace than that of RPWA-MIP and remains in the order of milliseconds (661 ms) while the runtime of RPWA-MIP exceeded the 2 days without providing a solution ($L = 15$ and $A = 12$). This proves that RPWA-D is much more scalable than RPWA-MIP.

C. Evaluation of RPWA-D

Given the non-scalability of RPWA-MIP, we focus in the following on studying the impact of varying network parameters on the deployment cost using RPWA-D while highlighting several trade-offs existing between the evaluated system parameters. Thus, unless stated otherwise, we consider a network consisting of $L = 10$ different locations, and a maximum of $M = 5$ edge servers that can be deployed in it. The maximum network delay is set to $1.5ms$.

1) Impact of varying the number of IoT applications for different industry verticals: We first investigate the impact of varying the number of available applications on the admitted workloads given IoT services belonging to different industry verticals [18], [19]. Thus, we consider four application types ($T = 4$) for each of the industry verticals depicted in Table IV and represented by their maximum allowable response time ($5ms \leq \delta_t \leq 110ms$). We evaluate the percentage of admitted workload for each of them as we vary the maximum number of applications that can be deployed in the network ($4 \leq A \leq 12$). Fig.3 illustrates an increase of the percentage of admitted workload with the increase of the number of applications. For instance, when considering Tactile Internet ($\delta = 5ms$), the admitted workload almost doubled when the number of applications went from $A = 4$ to $A = 8$. Similar observation can be deducted for the other presented types of industry verticals. This infers that, even though enough edge servers computing resources are available, a low number of applications was not able to admit all the workloads even when all the applications are deployed and assigned the maximum processing resources they can acquire (p_{max}^a). This shows that the maximum computing resources allocated to each application is not enough to meet the required QoS (response time), if assigned more workloads. In fact, an increase of the server delay will be observed with the increase in the workloads assigned to an application. Furthermore, Fig.3 depicts that when the response time increases (e.g., considering all the industry verticals), the percentage of admitted workload increases for a fixed number of applications. For instance, when $A = 12$, IoT applications utilizing Tactile Internet ($\delta_t = 5ms$) admitted 57.4% of the generated workloads while those used for factory automation ($\delta_t = 10ms$) were able to admit 94.3%. This trend continues

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2019.2894955, IEEE Transactions on Network and Service Management

as the response time increases to reach 100% of admission with the applications employed for Tele-Surgery. This can be explained by the fact that when the response time is relaxed, the workloads can tolerate higher queuing and processing delays which allows the applications to be able to meet the QoS requirement for a bigger fraction of their assigned load.

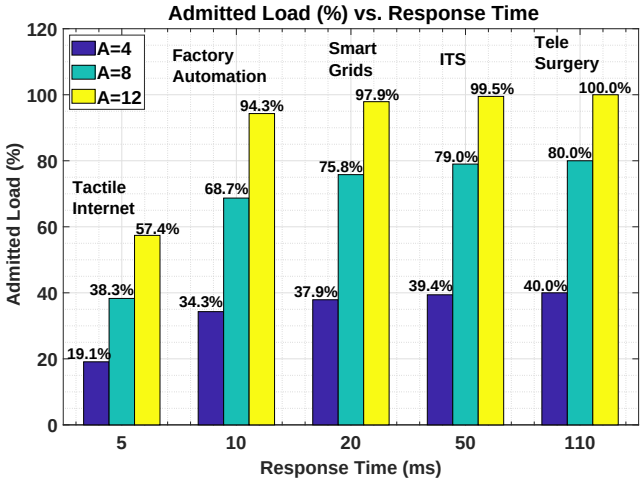


Fig. 3: Admission rate under varying number of applications.

2) Impact of varying workloads for different industry verticals: We evaluate the impact of the workload increase on the admission rate for different industry verticals. Thus, we consider $A = 12$ applications that can be deployed in the network and we vary the workload by varying the value of 205 requests/sec $\leq \lambda_l^t \leq$ 265 requests/sec. Fig.4 depicts that as the generated workloads per location per type increase for a given industry vertical, the percentage of admitted ones decreases, given that more workloads will be contending for the same computing resources (applications), which are not sufficient to serve all of them while meeting the required response time, since they will be experiencing higher queuing delay. However, this increased queuing delay can be tolerated if the response time increased. More precisely, one can note that as the workload for smart grid applications increased by 22.6% (from 205 requests/sec to 265 requests/sec), the admitted workload decreased by 6.4% showing that when all the computing resources (applications) are being consumed, the network fails to cope with the increased workloads and hence, less load is admitted. Further, when $\lambda_l^t = 265$ requests/sec, the admission rate increases with the increase of the response time to reach 95.5% for less latency sensitive applications such as those used in Tele-surgery.

3) Impact of varying the network size: We consider a factory automation industry where the allowed response time of its application is fixed to $\delta_t = 10ms$. We evaluate the trade-off existing between the number of locations and the number of applications in a network where at most $M = 5$ edge servers can be deployed. We account for $\lambda_l^t = 235$ requests/sec generated from each location for each of the $T = 4$ types of available applications.

Fig. 5 demonstrates that for the same number of applications, the admission rate decreases with the increase of the number of locations. This is explained by the fact that as the network size increases, more workloads are generated from the additional locations making the existing applications experience more congestion which leads to having their allocated computing resources fall short in serving the additional workloads within the allowed delay. However, as the number of applications increases (for a specific number of locations), more computing resources become available to process the additional workloads which explains the increase in the admission rate.

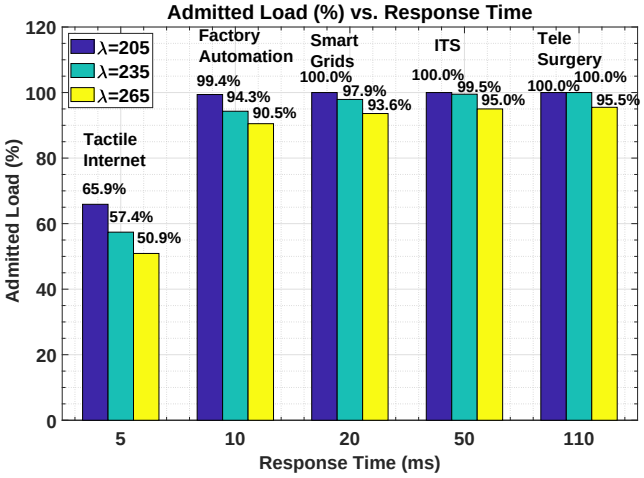


Fig. 4: Admission rate under varying workloads.

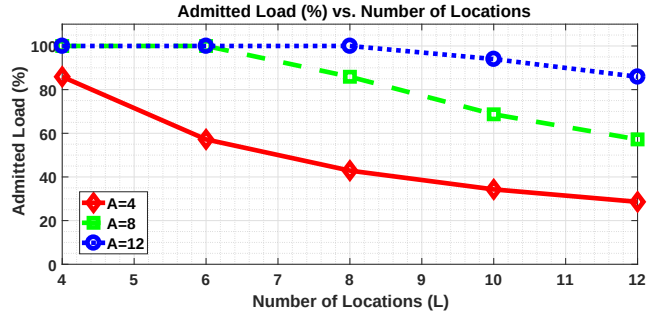


Fig. 5: Admitted rate over varying number of locations and applications.

4) Impact of varying the generated workloads and applications: Under the same simulation setup mentioned in the previous test (Section VI-C(3)), we provide a large scale test in order to demonstrate the feasibility of the RPWA-D approach. Thus, we increase the workload size for each type of application generated from every location by varying the value of λ_l^t between 150 requests/sec and 550 requests/sec. As we consider $L = 10$ locations and $T = 4$ types of applications, the total workload generated from all locations targeting all application types varied between 6000 requests/sec and 22000 requests/sec. In this test scenario, We vary the number of applications that can be placed on edge servers from $A = 4$ to $A = 12$ to study its impact on the admission rate. Fig. 6 illustrates that for the same number of applications, the admission rate decreases with the increase of the generated workload. This can be interpreted that the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2019.2894955, IEEE Transactions on Network and Service Management

provisioned processing capacities for a given number of applications cannot accommodate growing workload demand unless more resources are made available. As such, we can note that when the number of applications increases, the admission rate increases for the same amount of workload λ_l^t.

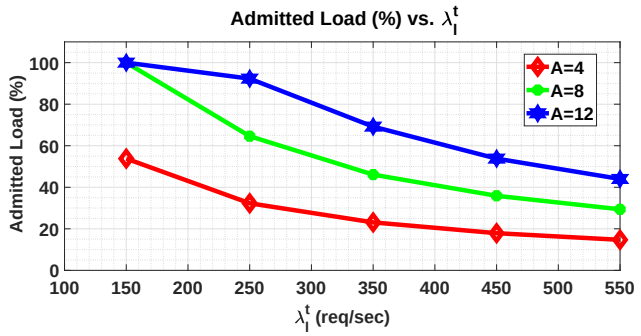


Fig. 6: Admission rate over varying workload and applications.

5) Impact of varying the edge servers capacities: We explore the relation between the edge servers capacities and the network utilization for different generated workloads. The network utilization is defined as the ratio between the total computing resources used by the deployed applications $\sum_{a\in A} p_a$ and the total computing capacity available by the deployed edge servers $\sum_{m\in M} c_m$ ($\frac{\sum_{a\in A} p_a}{\sum_{m\in M} c_m}$). Hence, we consider a factory automation industry ($\delta_t = 10ms$) of $L = 10$ locations, $M = 5$ edge servers that can be deployed to host up to $A = 12$ applications of $T = 4$ different types. Fig.7 depicts the percentage of network utilization as we increase the capacities of the edge servers (4Ghz $\leq c_m \leq$ 6.8GHz) and arrival rates of the requests ($\lambda_l^t \in \{150, 200\}$ requests/sec).

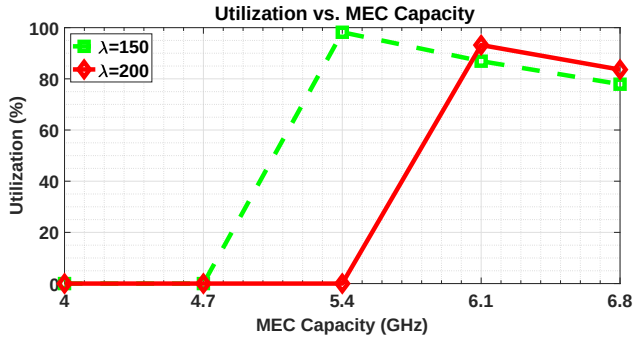


Fig. 7: Network utilization under variable MEC capacities.

Fig.7 delineates that 0% of the network resources are utilized when $c_m \in \{4, 4.7\}$ GHz and $\lambda_l^t \in \{150, 200\}$ requests/sec as RPWA-D fails to give a feasible solution since the total computing resources specified by the DALA-MIP and required to process the maximum amount of workload exceed those offered by all the edge servers that can be placed by the MESD-IP. The same behavior is observed when $c_m = 5.4$GHz and $\lambda_l^t = 200$ requests/sec. However, for a lower arrival rate of $\lambda_l^t = 150$ requests/sec, the network was fully utilized as the total computing resources provided by the edge servers were just enough to accommodate all the required applications determined by the DALA-MIP. As the value of c_m continues to increase, we observe that the network utilization decreases given that $\sum_{a\in A} p_a$ stabilized once all the workload is admitted and $\sum_{m\in M} c_m$ increases.

6) Deployment cost evaluation under varying network delay: We evaluate the impact of the increase of the maximum network delay on the edge servers deployment cost. Thus, we consider a smart grid system ($\delta_t = 20ms$) of $L = 10$ locations, $M = 10$ edge servers of varying configurations of computing capacity $c_m \in \{3, 5, 7, 9, 11\}$ GHz. We vary the setup cost between $\pi_l \in [5 - 20]$ unit cost and we set the cost of 1Ghz to be $k = 1$ unit cost. We consider $A = 20$ applications of $T = 4$ different types to process a workload generated at an arrival rate of $\lambda_l^t = 400$ requests/sec and vary the maximum network delay between $2ms$ and $10ms$.

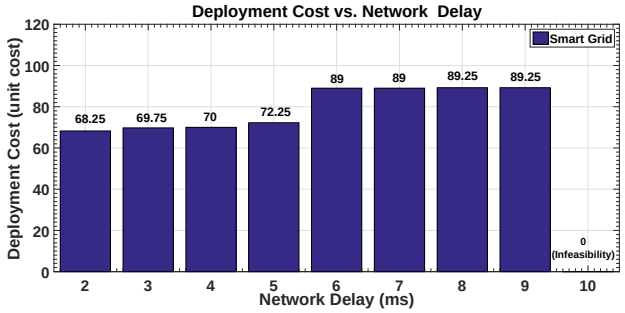


Fig. 8: Deployment cost under varying network delays.

In fact, increasing the network delay leads to very strict delay limit at the edge server to completely process requests by the hosted applications. As a result, more computing resources are required to be allocated to the applications when increased network delays are considered. Given that each of these applications has an upper bound on the maximum computing resources it can be assigned (p_{max}^a), more applications will be needed to process all the workload. This will require deploying more edge servers to be able to host these applications, which will eventually increase the deployment cost as shown in Fig.8. Note that, for a network delay of $10ms$, no solution was found as the maximum server delay remaining to process the workload within the response time is $0ms$ ($\delta_t' = \delta_t - 2d_n^{max} = 20 - 2 * 10 = 0ms$) (Section V). This, in fact, shows the importance of MEC in responding to the delay sensitive requirements of new emerging services.

7) Deployment cost evaluation under varying workload: Finally, we evaluate the impact of the increase of the generated workload on the deployment cost. We consider the same simulation setup described in Section VI-C (6) and we vary the workload arrival rate $\lambda_l^t \in [250 - 650]$ $requests/sec$. Our results depicted in Fig.9 show that the deployment cost increases with the increase of the workload as more edge servers need to be provisioned to handle such increase. However, one can note that the deployment cost remains the same for a load of 450, 550 and 650 requests/sec. With a more detailed evaluation of the cause of such result, we noticed that for all the aforementioned generated workloads, the total amount of admitted load remained stable at 17820 requests/sec with a deployment of all the $A = 20$ applications which explains

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2019.2894955, IEEE Transactions on Network and Service Management

that the number of available applications and the maximum processing capacity p_{max} they can be allocated limited the admission rate which stabilized the deployment cost.

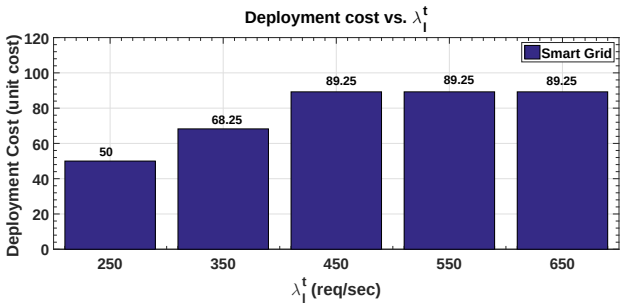


Fig. 9: Deployment cost under varying workload.

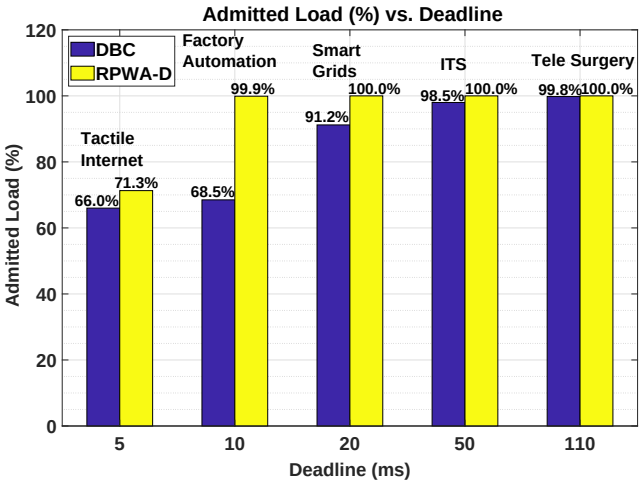


Fig. 10: Admitted rate over varying deadline.

D. Comparison of RPWA-D with existing work

As mentioned earlier, we are not aware of any existing literature that addressed the joint problem of MEC dimensioning, IoT application placement, and workload assignment. Moreover, most of the existing work did not account for various types of applications. As a result and to evaluate the efficiency of our proposed decomposition approach RPWA-D, we consider only one type of applications ($T = 1$) and compare against the work in [17] that addressed the placement of cloudlets (edge servers) and user (load) assignment in a Wireless Metropolitan Area Network WMAN. In order to apply their Density Based Clustering solution to our work, we use M/M/1 queue as the cloudlet model and assign deadlines to users' requests. Fig. 10 shows the simulation results of our proposed approach (RPWA-D) against the method proposed in [17] and referred to as DBC. Our simulation setup consists of $M = 5$ edge servers and $L = 10$ locations. We account for 150 users, each generating requests within the range of $[50 - 100]$ requests/sec. The value for λ_l^t in RPWA-D is calculated based on the total number of user requests generated from location l in the DBC algorithm. Our test regroups different vertical industries with varying response times $\{5, 10, 20, 50, 110\}ms$. As illustrated in Fig.10, using our proposed scheme RPWA-D, the admission rate can be increased by approximately 5.3%, 31.4%, 8.8%, 1.5% and 0.2% than the DBC algorithm for each of the presented vertical industries respectively. This is explained by the fact that DBC places cloudlets (edge servers) based on the most dense locations while in our approach an optimal edge servers placement is obtained. Further, in the DBC approach, a workload is either admitted or rejected completely, while, our approach allows a fraction of the load generated from one location to be admitted leading to more efficient and flexible workload assignment.

VII. Conclusion

In this paper, we studied the RPWA problem that jointly solves the MEC dimensioning, IoT applications placement and workload assignment sub-problems. To the best of our knowledge, this is the first work to address the three aforementioned sub-problems jointly. We mathematically formulated the RPWA problem as a MIP with the objective of minimizing the deployment cost while respecting the IoT applications' maximum allowable response times. Given the showed NP-Hardness of RPWA and the non scalability of the RPWA-MIP, we presented a decomposition approach (RPWA-D) to efficiently solve it. Unlike RPWA-MIP that considers admitting all the workloads, RPWA-D studies the percentage of the workloads that can be admitted given the available computing resources. Hence, RPWA-D can be used to study the trade-off between the increase of the workload admission rate and the extra computing resources needed to process the additional workload. Through extensive simulations, we verified the efficiency of RPWA-D under varying parameters and network conditions. Our proposed decomposition approach serves as a tool for network operators to develop cost effective strategies for edge network planning and design.

Appendix

A. Linearization of RPWA-MIP constraints

Linearization of Eq.(7)

Eq.(7) is non linear and can be linearized by declaring a new decision variable $\theta_a^{lm} \in \mathbb{R}^+$ such that:

$$\theta_a^{lm} = p_a y_a^{lm} \quad \begin{matrix}\forall l \in L \\ \forall m \in M \\ \forall a \in A\end{matrix} \tag{36}$$

Eq.(7) can then be replaced by the following equations:

$$\sum_{l \in L} \sum_{a \in A} \theta_a^{lm} \leq c_m \quad \forall m \in M \tag{37}$$

$$\theta_a^{lm} \leq H y_a^{lm} \quad \begin{matrix}\forall l \in L \\ \forall m \in M \\ \forall a \in A\end{matrix} \tag{38}$$

$$\theta_a^{lm} \leq p_a \quad \begin{matrix}\forall l \in L \\ \forall m \in M \\ \forall a \in A\end{matrix} \tag{39}$$

$$\theta_a^{lm} \geq p_a - (1 - y_a^{lm})H \quad \begin{matrix}\forall l \in L \\ \forall m \in M \\ \forall a \in A\end{matrix} \tag{40}$$

$$\theta_a^{lm} \geq 0 \quad \begin{matrix}\forall l \in L \\ \forall m \in M \\ \forall a \in A\end{matrix} \tag{41}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2019.2894955, IEEE Transactions on Network and Service Management

Where $H = p^a_{max}$ **Linarization of Eq.(18)**
The total delay in Eq.(18) is given by:

$$2\sum_{a\in A}\sum_{l'\neq l} h^{l'}_l z^a_{lt}\sigma^{l'}_a + \sum_{a\in A} z^a_{lt}\left(\frac{1}{\frac{p_a}{w_t} - \sum_{l''\in L} z^a_{l''t}\lambda^t_{l''}}\right) \leq \delta_t \quad \substack{\forall l\in L\\ \forall t\in T} \tag{42}$$

Eq.(42) is non linear and can be linearized by declaring a new decision variable ζ^a_{lt} such that:

$$\zeta^a_{lt} \geq z^a_{lt}\left(\frac{1}{\frac{p_a}{w_t} - \sum_{l''\in L} z^a_{l''t}\lambda^t_{l''}}\right) \quad \substack{\forall l\in L\\ \forall a\in A\\ \forall t\in T} \tag{43}$$

Hence, Eq.(42) becomes:

$$\underbrace{2\sum_{a\in A}\sum_{l'\neq l} h^{l'}_l z^a_{lt}\sigma^{l'}_a}_{d^{lt}_n} + \underbrace{\sum_{a\in A}\zeta^a_{lt}}_{d^{lt}_s} \leq \delta_t \quad \substack{\forall l\in L\\ \forall t\in T} \tag{44}$$

Eq.(44) is non linear and can be linearized by declaring a new decision variable $\psi^{al'}_{lt} \in \{0,1\}$ such that:

$$\psi^{al'}_{lt} = z^a_{lt}\sigma^{l'}_a \quad \substack{\forall l,l'\in L\\ \forall t\in T\\ \forall a\in A} \tag{45}$$

Eq.(44) can then be replaced by the following equations:

$$\underbrace{2\sum_{a\in A}\sum_{l'\neq l} h^{l'}_l \psi^{al'}_{lt}}_{d^{lt}_n} + \underbrace{\sum_{a\in A}\zeta^a_{lt}}_{d^{lt}_s} \leq \delta_t \quad \substack{\forall l\in L\\ \forall t\in T} \tag{46}$$

$$\psi^{al'}_{lt} \leq z^a_{lt} \quad \substack{\forall l,l'\in L\\ \forall t\in T\\ \forall a\in A} \tag{47}$$

$$\psi^{al'}_{lt} \leq \sigma^{l'}_a \quad \substack{\forall l,l'\in L\\ \forall t\in T\\ \forall a\in A} \tag{48}$$

$$\psi^{al'}_{lt} \geq z^a_{lt} + \sigma^{l'}_a - 1 \quad \substack{\forall l,l'\in L\\ \forall t\in T\\ \forall a\in A} \tag{49}$$

Eq.(43) is non linear and can be linearized by rewriting it as:

$$\zeta^a_{lt}\frac{p_a}{w_t} - \sum_{l''\in L}\zeta^a_{lt} z^a_{l''t}\lambda^t_{l''} \geq z^a_{lt} \quad \substack{\forall l\in L\\ \forall a\in A\\ \forall t\in T} \tag{50}$$

and then declaring a new decision variable $\beta^{al''}_{lt}$ such that:

$$\beta^{al''}_{lt} = \zeta^a_{lt} z^a_{l''t} \quad \substack{\forall l,l''\in L\\ \forall t\in T\\ \forall a\in A} \tag{51}$$

Hence, Eq.(50) can be replaced by the following:

$$\zeta^a_{lt}\frac{p_a}{w_t} - \sum_{l''\in L}\beta^{al''}_{lt}\lambda^t_{l''} \geq z^a_{lt} \quad \substack{\forall l\in L\\ \forall a\in A\\ \forall t\in T} \tag{52}$$

$$\beta^{al''}_{lt} \leq H z^a_{l''t} \quad \substack{\forall l,l'\in L\\ \forall t\in T\\ \forall a\in A} \tag{53}$$

$$\beta^{al'''}_{lt} \leq \zeta^a_{lt} \quad \substack{\forall l,l'\in L\\ \forall t\in T\\ \forall a\in A} \tag{54}$$

$$\beta^{al''}_{lt} \geq \zeta^a_{lt} - (1 - z^a_{l''t})H \quad \substack{\forall l,l'\in L\\ \forall t\in T\\ \forall a\in A} \tag{55}$$

$$\beta^{al''}_{lt} \geq 0 \quad \substack{\forall l,l'\in L\\ \forall t\in T\\ \forall a\in A} \tag{56}$$

Where H is a big integer number.
Eq.(52) is still non linear and can be linearized using the Mc-Cormick envelopes method. This method involves introducing a new decision variable γ^a_{lt} such that:

$$\gamma^a_{lt} = \zeta^a_{lt} p_a \quad \substack{\forall l\in L\\ \forall t\in T\\ \forall a\in A} \tag{57}$$

In addition, since ζ^a_{lt} and p_a are bounded by:

$$\because 0 \leq \zeta^a_{lt} \leq \frac{H}{p^a_{min}} \tag{58}$$

$$\because 0 \leq p_a \leq p^a_{max} \tag{59}$$

Thus, Eq.(52) can be replaced with the following equations:

$$\frac{\gamma^a_{lt}}{w_t} - \sum_{l''\in L}\beta^{al''}_{lt}\lambda^t_{l''} \geq z^a_{lt} \quad \substack{\forall l\in L\\ \forall a\in A\\ \forall t\in T} \tag{60}$$

$$\gamma^a_{lt} \geq 0 \quad \substack{\forall l\in L\\ \forall a\in A\\ \forall t\in T} \tag{61}$$

$$\gamma^a_{lt} \geq \frac{H}{p^a_{min}}p_a + \zeta^a_{lt}p^a_{max} - \frac{H}{p^a_{min}}p^a_{max} \quad \substack{\forall l\in L\\ \forall a\in A\\ \forall t\in T} \tag{62}$$

$$\gamma^a_{lt} \leq \frac{H}{p^a_{min}}p_a \quad \substack{\forall l\in L\\ \forall a\in A\\ \forall t\in T} \tag{63}$$

$$\gamma^a_{lt} \leq \zeta^a_{lt}p^a_{max} \quad \substack{\forall l\in L\\ \forall a\in A\\ \forall t\in T} \tag{64}$$

B. Linearization of DALA-MIP constraints

Linearization of Eq.(26)
Eq.(26) is non linear and can be rewritten as in Eq.(65)

$$\sum_{a\in A_t} z^a_r\frac{p_a}{w} - \sum_{a\in A_t}\sum_{r'\in R_t} z^a_r z^a_{r'}\lambda_{r'}\alpha_{r'} \geq \frac{1}{\delta'} \quad \forall r\in R_t \tag{65}$$

Eq.(65) can the be linearized by declaring three new decision variables: $\nu^a_r \in \mathbb{R}^+$ such that:

$$\nu^a_r = z^a_r p_a \quad \substack{\forall a\in A_t\\ \forall r\in R_t} \tag{66}$$

and $\tau^a_{rr'} \in \{0,1\}$ such that;

$$\tau^a_{rr'} = z^a_r z^a_{r'} \quad \substack{\forall a\in A_t\\ \forall r,r'\in R_t} \tag{67}$$

$\gamma^a_{rr'} \in [0-1]$ such that:

$$\gamma^a_{rr'} = \tau^a_{rr'}\alpha'_r \quad \substack{\forall a\in A_t\\ \forall r,r'\in R_t} \tag{68}$$

Eq.(65) can then be replaced by the following equations:

$$\nu^a_r \leq z^a_r p^a_{max} \quad \substack{\forall a\in A_t\\ \forall r\in R_t} \tag{69}$$

$$\nu^a_r \leq p_a \quad \substack{\forall a\in A_t\\ \forall r\in R_t} \tag{70}$$

$$\nu^a_r \geq p_a - p^a_{max}(1 - z^a_r) \quad \substack{\forall a\in A_t\\ \forall r\in R_t} \tag{71}$$

$$\tau^a_{rr'} \leq z^a_r \quad \substack{\forall a\in A_t\\ \forall r,r'\in R_t} \tag{72}$$

$$\tau^a_{rr'} \leq z^a_{r'} \quad \substack{\forall a\in A_t\\ \forall r,r'\in R_t} \tag{73}$$

$$\tau^a_{rr'} \geq z^a_r + z^a_{r'} - 1 \quad \substack{\forall a\in A_t\\ \forall r,r'\in R_t} \tag{74}$$

$$\gamma^a_{rr'} \leq \tau^a_{rr'} \quad \substack{\forall a\in A_t\\ \forall r,r'\in R_t} \tag{75}$$

$$\gamma^a_{rr'} \leq \alpha_r \quad \substack{\forall a\in A_t\\ \forall r,r'\in R_t} \tag{76}$$

$$\gamma^a_{rr'} \geq \alpha_r - (1 - \tau^a_{rr'}) \quad \begin{matrix}\forall a \in A_t \\ \forall r,r' \in R_t\end{matrix} \tag{77}$$

$$\sum_{a \in A_t} \frac{\nu^a_r}{w} - \sum_{a \in A_t} \sum_{r' \in R_t} \gamma^a_{rr'} \lambda_{r'} \geq \frac{1}{\delta'} \quad \forall r \in R_t \tag{78}$$

Linearization of Eq.(28)

Eq.(28) is non linear and can be linearized by declaring new decision variable $x^a_r \in [0,1]$ such that:

$$x^a_r = z^a_r \alpha_r \quad \begin{matrix}\forall a \in A_t \\ \forall r \in R_t\end{matrix} \tag{79}$$

Eq.(28) can then be replaced by the following equations:

$$x^a_r \leq z^a_r \quad \begin{matrix}\forall a \in A_t \\ \forall r \in R_t\end{matrix} \tag{80}$$

$$x^a_r \leq \alpha_a \quad \begin{matrix}\forall a \in A_t \\ \forall r \in R_t\end{matrix} \tag{81}$$

$$x^a_r \geq \alpha_r - (1 - z^a_r) \quad \begin{matrix}\forall a \in A_t \\ \forall r \in R_t\end{matrix} \tag{82}$$

$$\frac{p_a}{w} - \sum_{r \in R_t} x^a_r \lambda_r > 0 \quad \forall a \in A_t \tag{83}$$

REFERENCES

[1] Raymond James et al. The internet of things: a study in hype, reality, disruption, and growth. *Raymond James US Research, Technology & Communications, Industry Report*, 2014.

[2] In Lee and Kyoochun Lee. The internet of things (iot): Applications, investments, and challenges for enterprises. *Business Horizons*, 58(4):431–440, 2015.

[3] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things: A survey. *Computer networks*, 54(15):2787–2805, 2010.

[4] Shanhe Yi, Cheng Li, and Qun Li. A survey of fog computing: concepts, applications and issues. In *Proceedings of the 2015 Workshop on Mobile Big Data*, pages 37–42. ACM, 2015.

[5] Eduardo Cuervo, Aruna Balasubramanian, Dae-ki Cho, Alec Wolman, Stefan Saroiu, Ranveer Chandra, and Paramvir Bahl. Maui: making smartphones last longer with code offload. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*, pages 49–62. ACM, 2010.

[6] Xu Chen, Lei Jiao, Wenzhong Li, and Xiaoming Fu. Efficient multi-user computation offloading for mobile-edge cloud computing. *IEEE/ACM Transactions on Networking*, 24(5):2795–2808, 2016.

[7] Ruozhou Yu, Guoliang Xue, and Xiang Zhang. Application provisioning in fog computing-enabled internet-of-things: A network perspective. *IEEE International Conference on Computer Communications (INFOCOM)*, 2018.

[8] Qiang Fan and Nirwan Ansari. Application aware workload allocation for edge computing based iot. *IEEE Internet of Things Journal*, 2018.

[9] ETSI. Mobile-edge computing introductory technical white paper, (2014).

[10] Koustabh Dolui and Soumya Kanti Datta. Comparison of edge computing implementations: Fog computing, cloudlet and mobile edge computing. In *Global Internet of Things Summit (GIoTS), 2017*, pages 1–6. IEEE, 2017.

[11] Tarik Taleb, Konstantinos Samdanis, Badr Mada, Hannu Flinck, Sunny Dutta, and Dario Sabella. On multi-access edge computing: A survey of the emerging 5g network edge cloud architecture and orchestration. *IEEE Communications Surveys & Tutorials*, 19(3):1657–1681, 2017.

[12] Pavel Mach and Zdenek Becvar. Mobile edge computing: A survey on architecture and computation offloading. *IEEE Communications Surveys & Tutorials*, 19(3):1628–1656, 2017.

[13] Shuo Wang, Xing Zhang, Yan Zhang, Lin Wang, Juwo Yang, and Wenbo Wang. A survey on mobile edge networks: Convergence of computing, caching and communications. *IEEE Access*, 5:6757–6779, 2017.

[14] Arslan Munir, Prasanna Kansakar, and Samee U Khan. Ifciot: Integrated fog cloud iot: A novel architectural paradigm for the future internet of things. *IEEE Consumer Electronics Magazine*, 6(3):74–82, 2017.

[15] Bhaskar Prasad Rimal, Dung Pham Van, and Martin Maier. Mobile-edge computing versus centralized cloud computing over a converged fiwi access network. *IEEE Transactions on Network and Service Management*, 14(3):498–513, 2017.

[16] Mike Jia, Weifa Liang, and Zichuan Xu. Qos-aware task offloading in distributed cloudlets with virtual network function services. In *Proceedings of the 20th ACM International Conference on Modelling, Analysis and Simulation of Wireless and Mobile Systems*, pages 109–116. ACM, 2017.

[17] Mike Jia, Jiannong Cao, and Weifa Liang. Optimal cloudlet placement and user to cloudlet allocation in wireless metropolitan area networks. *IEEE Transactions on Cloud Computing*, 2017.

[18] M Maternia, S Eddine El Ayoubi, M Fallgren, P Spapis, Y Qi, D Martín-Sacristán, Óscar Carrasco, M Fresia, M Payaró, M Schubert, et al. 5g ppp use cases and performance evaluation models. *see https://5g-ppp. eu/wp-content/uploads/2014/02/5G-PPP-use-cases-and-performance-evaluation-modeling_v1. 0. pdf*, 2016.

[19] 3GPP TR 22.864. Feasibility study on new services and markets technology enablers - network operation - stage 1.

[20] Pawani Porambage, Jude Okwuibe, Madhusanka Liyanage, Mika Ylianttila, and Tarik Taleb. Survey on multi-access edge computing for internet of things realization. *arXiv preprint arXiv:1805.06695*, 2018.

[21] Lei Zhao, Wen Sun, Yongpeng Shi, and Jiajia Liu. Optimal placement of cloudlets for access delay minimization in sdn-based internet of things networks. *IEEE Internet of Things Journal*, 2018.

[22] Alberto Ceselli, Marco Premoli, and Stefano Secci. Mobile edge cloud network design optimization. *IEEE/ACM Transactions on Networking*, 25(3):1818–1831, 2017.

[23] Xavier Dos Santos, José Rafael, Tim Wauters, Bruno Volckaert, and Filip De Turck. Resource provisioning for iot application services in smart cities. In *CNSM2017, the 13e International Conference on Network and Service Management*, pages 1–9, 2017.

[24] Mohammed Islam Naas, Laurent Lemarchand, Jalil Boukhobza, and Philippe Raipin. A graph partitioning-based heuristic for runtime iot data placement strategies in a fog infrastructure. In *SAC 2018: Symposium on Applied Computing*, 2018.

[25] Min Chen and Yixue Hao. Task offloading for mobile edge computing in software defined ultra-dense network. *IEEE Journal on Selected Areas in Communications*, 36(3):587–597, 2018.

[26] Sivadon Chaisiri, Bu-Sung Lee, and Dusit Niyato. Optimization of resource provisioning cost in cloud computing. *IEEE Transactions on Services Computing*, 5(2):164–177, 2012.

[27] Yeongho Choi and Yujin Lim. Optimization approach for resource allocation on cloud computing for iot. *International Journal of Distributed Sensor Networks*, 2016:23, 2016.

[28] I Bolodurina and D Parfenov. Development and research of models of organization distributed cloud computing based on the software-defined infrastructure. *Procedia Computer Science*, 103:569–576, 2017.

[29] Deze Zeng, Lin Gu, Song Guo, Zixue Cheng, and Shui Yu. Joint optimization of task scheduling and image placement in fog computing supported software-defined embedded system. *IEEE Transactions on Computers*, 65(12):3702–3712, 2016.

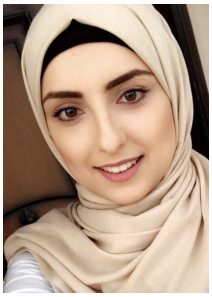
[30] Ling-Yun Wu, Xiang-Sun Zhang, and Ju-Liang Zhang. Capacitated facility location problem with general setup cost. *Computers & Operations Research*, 33(5):1226–1241, 2006.

[31] Sven O Krumke and Clemens Thielen. The generalized assignment problem with minimum quantities. *European Journal of Operational Research*, 228(1):46–55, 2013.

[32] Maria A Lema, Andres Laya, Toktam Mahmoodi, Maria Cuevas, Joachim Sachs, Jan Markendahl, and Mischa Dohler. Business case and technology analysis for 5g low latency applications. *IEEE Access*, 5:5917–5935, 2017.

[33] Philipp Schulz, Maximilian Matthe, Henrik Klessig, Meryem Simsek, Gerhard Fettweis, Junaid Ansari, Shehzad Ali Ashraf, Bjoern Almeroth, Jens Voigt, Ines Riedel, et al. Latency critical iot applications in 5g: Perspective on the design of radio interface and network architecture. *IEEE Communications Magazine*, 55(2):70–78, 2017.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2019.2894955, IEEE Transactions on Network and Service Management



Nouha Kherraf received her Bachelor degree in Computer Engineering from Qatar University, Doha, Qatar in 2015. She is currently pursuing her Master of science degree in Electrical and Computer Engineering at Concordia University, Montreal, Canada. She held the position of a Research Assistant in the Electrical and Computer Engineering department at Texas A&M University at Qatar in 2016 and 2017. Her current research interests include edge computing, 5G, Internet of Things and network optimization.

Hyame A. Alameddine received her Computer Engineering degree in information, systems and multimedia from CNAM University-Paris-France in 2015. She is currently pursuing her Ph.D. degree at the Concordia Institute of Information Systems Engineering, Concordia University-Montreal-Canada. She is a co-founder of the Montreal Operations Research student chapter and occupied the positions of Academic Events Director in the past year and is currently the Vice President. Her research interests are in the areas of cloud computing, network function virtualization, software defined networks, edge computing, Internet of Things, 5G and network optimization.

Sanaa Sharafeddine is an Associate Professor of Computer Science at the Lebanese American University. She received her doctoral degree in Communications Engineering from Munich University of Technology (TUM) in 2005 in collaboration with Siemens AG research labs in Munich. She received L'Oreal-UNESCO's International Rising Talent Award in 2015 and Pan-Arab Regional Fellowship Award in 2013. She is an Area Editor for Elsevier Ad Hoc Networks and Associate Editor for IEEE Access. Her research interests include UAV-assisted solutions in 5G, wireless networking, and multimedia services.

Chadi M. Assi received the Ph.D. degree from the City University of New York (CUNY) in 2003. He is currently a Full Professor at Concordia University. He was a recipient of the Prestigious Mina Rees Dissertation Award from CUNY in 2002 for his research on wavelength-division multiplexing optical networks. He is on the Editorial Board of IEEE Communications Surveys and Tutorials, the IEEE Transactions on Communications, and the IEEE Transactions on Vehicular Technologies. His current research interests are in the areas of network design and optimization, network modelling, and network reliability.

Ali Ghrayeb received the Ph.D. degree in electrical engineering from The University of Arizona, Tucson, AZ, USA, in 2000. He is currently a Professor with the Department of Electrical and Computer Engineering, Texas A&M University at Qatar. Prior to his current position, he was a professor with the Department of Electrical and Computer Engineering, Concordia University, Montreal, Canada. His research interests include wireless and mobile communications, physical layer security, massive MIMO, and visible light communications. He served as an instructor or co-instructor in technical tutorials at several major IEEE conferences. He served as the Executive Chair for the 2016 IEEE WCNC conference. He has served on the editorial board of several IEEE and non-IEEE journals. He is a Fellow of the IEEE.