# Resource Provisioning for IoT application services in Smart Cities

José Santos, Tim Wauters, Bruno Volckaert and Filip De Turck
Ghent University - imec, IDLab, Department of Information Technology
Technologiepark-Zwijnaarde 15, 9052 Gent, Belgium
Email: josepedro.pereiradossantos@ugent.be

*Abstract*— **In the last years, traffic over wireless networks has been increasing exponentially, due to the impact of Internet of Things (IoT) and Smart Cities. Current networks must adapt to and cope with the specific requirements of IoT applications since resources can be requested on-demand simultaneously by multiple devices on different locations. One of these requirements is low latency, since even a small delay for an IoT application such as health monitoring or emergency service can drastically impact their performance. To deal with this limitation, the Fog computing paradigm has been introduced, placing cloud resources on the edges of the network to decrease the latency. However, deciding which edge cloud location and which physical hardware will be used to allocate a specific resource related to an IoT application is not an easy task. Therefore, in this paper, an Integer Linear Programming (ILP) formulation for the IoT application service placement problem is proposed, which considers multiple optimization objectives such as low latency and energy efficiency. Solutions for the resource provisioning of IoT applications within the scope of Antwerp's City of Things testbed have been obtained. The result of this work can serve as a benchmark in future research related to placement issues of IoT application services in Fog Computing environments since the model approach is generic and applies to a wide range of IoT applications.**

## I. Introduction

In recent years, the Internet of Things (IoT) has introduced a whole new set of challenges and opportunities by transforming objects of everyday life in communicating devices [1]. Moreover, with the advent of the IoT, the concept of Smart City has become even more popular in the last few years [2]. Smart City applications will transform a wide range of services in different domains of urban life, for instance, by creating intelligent smart grid networks, improving public transportation, developing smart car parking and real-time industrial automation applications and reducing traffic congestion. Essentially, millions of devices will be connected to the network, sending and receiving data to the cloud, which current networks will not be able to support [3]. Therefore, it is necessary to adapt existing cloud and network architectures to future needs and design and develop new management functionalities to help meet the strict requirements of future Smart City IoT applications.

Fog Computing extends the Cloud Computing paradigm by bringing cloud services closer to the end devices, thus reducing the communication latency [4], [5]. However, there is still a large number of research challenges associated with this approach since Fog Computing is in its early stages and needs more time to evolve. One of the main challenges is the proper resource allocation, since services can be placed in a highly congested location, or even further from the end devices, which would result in a higher communication latency because current end devices and gateways are lacking in terms of processing power, storage capacity and memory [6]. Moreover, few resource management strategies are currently addressing the real-time constraints of Smart City IoT applications while minimizing resource costs and maximizing quality of service (QoS). Therefore, efficient resource allocation strategies are needed in order to address all these issues.

This paper presents an Integer Linear Programming (ILP) formulation for the IoT application service placement problem in order to evaluate resource provisioning in Smart City scenarios. IoT applications have been considered as a set of multiple communicating services, like applications designed in Service-Oriented Architectures (SOA). SOA-based architectures have been used in the last years for IoT [7], [8]. This way, an IoT application can be designed as a coordinated workflow of multiple services which are associated with actions performed by end devices. Research have been carried out to solve the issues of abstracting end device functionalities, trying to provide a suitable architecture with service management and composition capabilities able to link a set of common services in a set of IoT applications. This proposed architecture is presented in Figure 1. Each communicating service can be provided by a virtual machine which may be used by multiple tenants. In a Smart City scenario, when there is a request from an IoT application, resources should be distributed within the network ensuring that the services composing the IoT application are allocated and instantiated close to the end device that made the request. Multiple factors should be taken into account to ensure proper resource allocation such as latency, energy efficiency, bandwidth and cost.

The remainder of the paper is organized as follows. In the next Section, related work is discussed. Section III introduces the proposed ILP model for the resource provisioning of IoT application services. In Section IV, evaluation scenario is described which is followed by the evaluation results in Section V. Finally, conclusions are presented in Section VI.

## II. Related Work

In recent years, studies have been carried out in order to deal with application placement issues in IoT. In [9], a

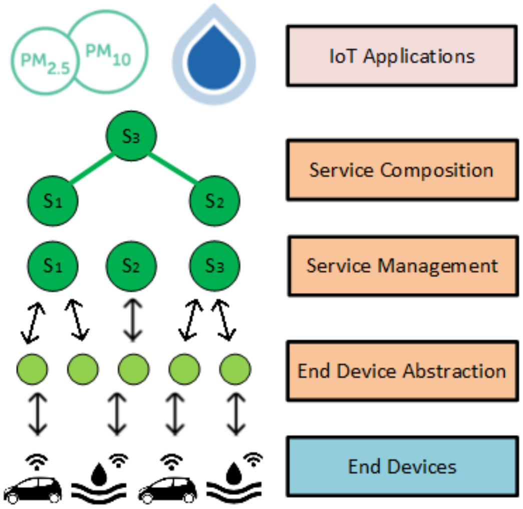

Fig. 1. SOA-based architecture for IoT applications.

model and an architecture have been introduced to deal with resource provisioning in fog computing environments focusing on the reduction of service latency for IoT applications. In [10], a energy management strategy for a Fog Computing platform is presented. Moreover, SmartSantander [11] worked on a suitable architectural model for the IoT and the inherent challenges of service provisioning in Smart Cities was in their scope [12]. In [13], a resilient IoT architecture for Smart Cities has been presented and in [14] the remaining issues of integrating Cloud Computing and IoT are discussed, where the integration was referred to as Cloud of Things. In [15], a novel scheme for an energy efficient IoT based on Wireless Sensor Networks (WSN) has been introduced focusing on WSN characteristics. Cloud requirements have not been included on the model.

In recent years, research efforts have been carried out to overcome application placement issues mainly focused on cloud environments where IoT or Smart Cities contexts have not been considered. Many works focused only on the allocation of virtual network functions (VNFs) or Virtual machines (VMs) on clouds [16], [17]. However, recently in [18], a resource aware placement algorithm of IoT applications in Fog Computing environments has been presented focusing on latency, network usage and energy consumption. Only static network topologies have been evaluated and no wireless constraint has been introduced. Nevertheless, a lot of challenges still remain to fully address resource provisioning of Smart City IoT applications, since previous research does not take into account requirements stemming from the characteristics of wireless networks.

This way, in this paper, a resource provisioning ILP model is presented that goes beyond the current state-of-the-art by taking into account not only cloud requirements but also wireless constraints which were, to the best of our knowledge, not yet explored in-depth in literature.

## III. The ILP model

### A. Model Description

The resource provisioning model considers cloud and wireless characteristics. The cloud model is based on the previous work done by Moens et al. [16] on network-aware placement of service oriented applications in clouds. Regarding wireless characteristics, a 802.11ah [19] Low-Power Wide-Area Network (LPWAN) has been modeled as an ILP formulation. An IoT application is composed of multiple communicating services. End devices send requests for these IoT applications through wireless gateways. These gateways communicate with the fog-cloud infrastructure, managing a set of computational resources. Each service must be allocated and instantiated on a given set of computational resources, subject to multiple constraints [16]:

- Computational resources have limited CPU and memory.
- Communication links between computational resources have limited bandwidth.
- Gateways have limited association identifiers (AIDs) so end devices can associate and send requests for IoT applications.
- IoT application services cannot be instantiated on every computational resource, due to specific hardware or software requirements.

The work in [16] incorporates multiple optimization objectives which have been extended to address the IoT application placement problem identified in this paper. This way, the model is executed iteratively so that in each iteration a different optimization objective is considered. To retain the objective values obtained in the previous iterations, additional constraints are added to the model. Thus, the solution space continuously decreases since iterations must satisfy the previous optimal solutions. Every iteration refines the previous obtained solution by improving the model with an additional optimization objective. The optimization objectives considered in the model are the following:

1) Maximization of accepted IoT application requests.
2) Maximization of service bandwidth.
3) Minimization of service migrations between iterations.
4) Minimization of number of active computational nodes.
5) Minimization of the number of active gateways.
6) Minimization of hop count between computational nodes and end devices.
7) Minimization of path loss.

Optimization objectives from 1) to 4) have been already considered in [16]. The work has been extended with three additional optimization objectives, from 5) to 7).

### B. Variables

Inputs and decision variables used in the model are shown in Table I and in Table II, respectively. A set of applications $A$ composed of communicating services $S$ are given. The number of requests and the total number of requests for an application $a \,\varepsilon\, A$ are given by $R_a$ and $D_a$ respectively. A binary request matrix $\Phi_{a,r,ed}$ indicates if an end device $ed \,\varepsilon\, N_{ed}$ made the

TABLE I
THE INPUT VARIABLES OF THE ILP MODEL

| Symbol | Description |
|---|---|
| $N_c$ | The set of computational nodes on which services are executed. |
| $N_f$ | The set of fog clouds on the network which manage the computational nodes. |
| $N_{gw}$ | The set of wireless gateways on the network. |
| $N_{ed}$ | The set of end devices on the network. |
| $A$ | The set of all IoT applications. Each IoT application is composed of a set of communicating services. |
| $S$ | The set of all communicating services. |
| $R_a$ | The set containing all requests for an application $a \varepsilon A$. |
| $D_a$ | The total number of requests for an application $a \varepsilon A$. |
| $R_{ed}$ | A binary value that indicates if the end device $ed$ sent a request for an IoT application $a \varepsilon A$. |
| $\Omega_n$ | The total CPU capacity (in GHz) of the computational node $n \varepsilon N_c$. |
| $\Gamma_n$ | The total memory capacity (in GB) of the computational node $n \varepsilon N_c$. |
| $\omega_s$ | The CPU requirement (in GHz) of the service $s \varepsilon S$. |
| $\gamma_s$ | The memory requirement (in GB) of the service $s \varepsilon S$. |
| $\Theta_{gw}$ | The total association identifiers available on a gateway $gw \varepsilon N_{gw}$. |
| $\theta_{ed}$ | Each end device $ed \varepsilon N_{ed}$ needs an association identifier to associate with a gateway. |
| $L$ | Set of locations where IoT application requests are generated by end devices. |
| $\Phi_{a,r,ed}$ | The Request matrix. If $\Phi_{a,r,ed} = 1$, the end device $ed$ made the $r$th request of application $a$. |
| $R_{s,n}$ | The Relation matrix. If $R_{s,n} = 1$, the communicating service $s$ can be allocated on node $n$. If $R_{s,n} = 0$, the communicating service $s$ cannot be instantiated on node $n$. |
| $I_{a,s}$ | The Instance matrix. If $I_{a,s} = 1$, the communicating service $s$ is part of application $a$. If $I_{a,s} = 0$, the communicating service $s$ is not part of application $a$. |
| $H_{n,ed}$ | The Hop Count matrix between computational nodes and end devices indicates the number of devices between the computational node $n$ and the end device $ed$. |
| $B_{n_1,n_2}$ | The Bandwidth matrix between computational nodes indicates the bandwidth (Mbit/s) available between the computational node $n_1$ and the node $n_2$. |
| $C_{n_1,n_2}$ | The Communication matrix between computational nodes indicates the bandwidth (Mbit/s) required between services of an IoT application. |
| $D_{gw,ed}$ | The Distance matrix between gateways and end devices indicates the distance (in meters) between a gateway and an end device. |
| $PL_{gw,ed}$ | The Path Loss matrix between gateways and end devices indicates the path loss (in dB) between a gateway and an end device. |
| $A_{ed,gw}$ | The Association matrix. If $A_{ed,gw} = 1$, the end device $ed$ can associate with the gateway $gw$. If $A_{ed,gw} = 0$, the end device $ed$ cannot associate with the gateway $gw$. |
| $E_{n,l}$ | $E_{n,l} = 1$ indicates that computational node $n$ is at location $l$. |
| $E_{f,l}$ | $E_{f,l} = 1$ indicates that fog cloud $f$ is at location $l$. |
| $E_{gw,l}$ | $E_{gw,l} = 1$ indicates that gateway $gw$ is at location $l$. |
| $E_{ed,l}$ | $E_{ed,l} = 1$ indicates that end device $ed$ is at location $l$. |
| $E_{n,f}$ | $E_{n,f} = 1$ indicates that computational node $n$ is managed by fog cloud $f$. |

$r$th request of application $a$. Also, a binary instance matrix $I$ indicates if a service $s \varepsilon S$ is part of an application $a \varepsilon A$. Each service $s$ has a CPU and a memory requirement represented by $\omega_s$ (in GHz) and $\gamma_s$ (in GB) respectively. The communicating services must be allocated on computational nodes $n \varepsilon N_c$. Each computational node $n$ has a CPU and a memory capacity represented by $\Omega_n$ (in GHz) and $\Gamma_n$ (in GB), respectively.

TABLE II
THE DECISION VARIABLES OF THE ILP MODEL

| Symbol | Description |
|---|---|
| $G_{a,r}$ | The acceptance matrix. If $G_{a,r} = 1$, the $r$th request of IoT application $a$ can be accepted. If $G_{a,r} = 0$, the $r$th request of IoT application $a$ cannot be accepted. |
| $P^{a,r}_{s,n}$ | The placement matrix. If $P^{a,r}_{s,n} = 1$, an instance of service $s$ is executed on computational node $n$ for the $r$th request of IoT application $a$. |
| $U_{s,n}$ | The service execution matrix. If $U_{s,n} = 1$, an instance of service $s$ is allocated on computational node $n$. If $U_{s,n} = 0$, there is not an instance of service $s$ allocated on computational node $n$. |
| $U_{ed,gw}$ | The end device execution matrix. If $U_{ed,gw} = 1$, the end device $ed$ is associated with gateway $gw$. |
| $U_{gw}$ | The gateway utilization matrix. $U_{gw} = 1$ indicates that there is at least one end device associated with gateway $gw$. |
| $U_n$ | The computational node utilization matrix. $U_n = 1$ indicates that there is at least one service allocated on computational node $n$. |
| $F^{a,r}_{s_1,s_2}(n_1,n_2)$ | The flow matrix contains the bandwidth (in Mbit/s) belonging to the $r$th request of IoT application $a$ that is used in the communication between services $s_1$ and $s_2$ which are allocated on node $n_1$ and $n_2$, respectively. |
| $z^{a,r}_{s_1,s_2}$ | The service bandwidth matrix contains the amount of bandwidth for every flow in the communication between services $s_1$ and $s_2$ for the $r$th request of IoT application $a$. |

A binary relation matrix $R$ is used to indicate if an instance of service $s$ could be allocated on a given computational node $n \varepsilon N_c$. If $R_{s,n} = 1$, the communicating service $s$ can be allocated on node $n$. Otherwise, due to software or hardware limitations, the service $s$ cannot be instantiated on node $n$. Moreover, a binary acceptance matrix $G$ is used to indicate if the $r$th request of application $a$ can be accepted. If $G_{a,r} = 1$, all the services that compose application $a$ are allocated on computational nodes $n \varepsilon N_c$ and therefore the $r$th request of application $a$ is accepted. A binary placement matrix $P$ is used to represent in which computational node $n$ an instance of a service $s$ is allocated. If $P^{a,r}_{s,n} = 1$, an instance of service $s$ is executed on the computational node $n$ for the $r$th request of the IoT application $a$. A set of locations $L$ is used to define where IoT applications requests are generated. Multiple binary matrices $E$ are considered to define in which location fog clouds $f \varepsilon N_f$, end devices $ed \varepsilon N_{ed}$, gateways $gw \varepsilon N_{gw}$ and computational nodes $n \varepsilon N_c$ are on the network. One additional binary matrix $E$ is considered to indicate if a computational node $n$ is managed by a fog cloud $f$.

Regarding wireless formulation, the total AIDs available on a given gateway $gw \varepsilon N_{gw}$ is given by $\Theta_{gw}$. Each end device $ed \varepsilon N_{ed}$ needs an AID to associate with a gateway which is represented by $\theta_{ed}$. Moreover, a distance matrix $D$ indicates the distance (in meters) between a gateway $gw \varepsilon N_{gw}$ and an end device $ed \varepsilon N_{ed}$ while a path loss matrix $PL$ indicates the path loss (in dB) between a gateway $gw \varepsilon N_{gw}$ and an end device $ed \varepsilon N_{ed}$. A $R_{ed}$ binary variable indicates if an end device $ed$ sent requests for an IoT application. If $R_{ed} = 1$, the end device $ed$ sent a request for an IoT application $a \varepsilon A$.

Otherwise, $R_{ed} = 0$. An additional binary association matrix $A$ is used to indicate if an end device $ed$ can associate with a gateway $gw$. This association is based on the distance matrix $D$. If $D_{gw,ed}$ is less than one thousand meters, the end device $ed$ can associate with gateway $gw$ and then $A_{ed,gw} = 1$. Otherwise, $A_{ed,gw} = 0$. The limit is set to one thousand meters because this is the maximum coverage range in 802.11ah networks [20].

A hop count matrix $H$ indicates the number of devices between the computational node $n$ and the end device $ed$. Moreover, additional decision execution and utilization matrices $U$ are considered. First, service execution matrix $U_{s,n}$ and end device execution matrix $U_{ed,gw}$ indicate if a service instance $s$ is allocated on computational node $n$ and if an end device $ed$ is associated with the gateway $gw$, respectively. Secondly, computational node utilization matrix $U_n$ and gateway utilization matrix $U_{gw}$ indicate if there is at least a service running on computational node $n$ and if there is at least an end device associated with gateway $gw$, respectively.

Then, as mentioned by Moens et al. [16], a two-stage approach has been considered to allocate the bandwidth between communication services $s_1$ and $s_2$ that are part of the same IoT application request $r$. A bandwidth matrix $B$ which indicates the available bandwidth (in Mbit/s) between two computational nodes is considered. Moreover, a communication matrix $C$ is defined, where $C_{s_1,s_2}$ indicates the needed bandwidth (in Mbit/s) between two communication services. The flow bandwidth between two computational nodes is defined by the flow matrix $F$. $F^{a,r}_{s_1,s_2}(n_1, n_2)$ contains the bandwidth (in Mbit/s) belonging to the $r$th request of IoT application $a$ that is used in the communication between services $s_1$ and $s_2$ which are allocated on node $n_1$ and $n_2$, respectively. Finally, $z^{a,r}_{s_1,s_2}$ is a decision variable that indicates the percentage of the requested bandwidth between services $s_1$ and $s_2$ that is guaranteed for the $r$th request of the IoT application $a$.

Each optimization objective is detailed bellow.

### *C. Maximizing the Number of Accepted Requests - MAX R*

The goal of this optimization is to maximize the number of accepted requests on the network. This objective can be represented as shown in (1). This optimization objective is subjected to multiple constraints. Constraints presented in [16] have been considered in the model, which has been extended with additional ones related to the wireless formulation.

$$max \sum_{a\,\varepsilon\,A} \sum_{r\,\varepsilon\,R_a} G_{a,r} \times \left( \sum_{s\,\varepsilon\,S} I_{a,s} \times \omega_s \right) \tag{1}$$

In 802.11ah networks, end devices associate with gateways through an AID, a unique value assigned to an end device by the gateway during association handshake [21]. A gateway cannot have more than 8191 associated stations according to the latest standard. However, the association limitation has been set to 50 since an urban macro deployment with extended range has been considered [21]. This way, with this lower limitation, it has been assumed that good channel conditions are always achieved and that all requests sent for IoT applications can be accepted. Therefore, a constraint must be added to the model ensuring that the AIDs limit in each gateway is respected. This way, by using the end device execution matrix $U$, the AIDs limitation can be expressed as shown in (2). The total amount of AIDs attributed in a gateway must be less than the total amount of AIDs available.

$$\forall gw\,\varepsilon\,N_{gw} : \sum_{ed\,\varepsilon\,N_{ed}} \theta_{ed} \times U_{ed,gw} \leq \Theta_{gw} \tag{2}$$

Secondly, a constraint is added to ensure that end devices are associated with one gateway to be able to send requests for IoT applications. This constraint is represented by (3).

$$\forall ed\,\varepsilon\,N_{ed} : \sum_{gw\,\varepsilon\,N_{gw}} U_{ed,gw} \times A_{ed,gw} = 1 \tag{3}$$

### *D. Maximizing the Satisfied Service Bandwidth Demand - MAX SB*

On the previous objective, an IoT application is allocated on the network if at least 80% of the required bandwidth is guaranteed. The goal of this optimization is to further increase the allocated bandwidth, ensuring that the maximum capacity available is allocated to the communicating services that compose the IoT applications requested on the network. This maximization is expressed in (4).

$$max \sum_{a\,\varepsilon\,A} \sum_{r\,\varepsilon\,R_a} \sum_{s_1,s_2\,\varepsilon\,S} z^{a,r}_{s_1,s_2} \tag{4}$$

### *E. Minimizing Service Migrations On Subsequent Iterations - MIN M*

The goal of this optimization is to minimize service migrations between subsequent iterations of the model. Since the model is executed iteratively, the execution matrix from the previous iteration $U^{i-1}$ is added to the model, which is used to compare with the current execution matrix $U$ in order to reduce the service migrations needed to achieve the next optimization objective. Therefore, the minimization of service migrations can be expressed as shown in (5). Services may have to be migrated from one computational node to another in order to find the optimal solution. However, it might be preferable to find a solution where service reallocations are minimized so delay caused by reallocations is kept at a minimum.

$$min \sum_{s\,\varepsilon\,S} \sum_{n\,\varepsilon\,N_c} \mid U_{s,n} - U^{i-1}_{s,n} \mid \tag{5}$$

### *F. Minimizing the Number of Active Computational Nodes - MIN $N_c$*

The goal of this optimization is to minimize the number of active computational nodes in the network, which results in cost and energy savings. By using the computational node utilization decision variable $U_n$, the minimization can be expressed as shown in (6).

$$min \sum_{n \, \varepsilon \, N_c} U_n \tag{6}$$

The binary decision variable $U_n$ is subject to additional constraints, ensuring that it only takes on value 0 if there is no communicating service allocated on that computational node [16]. This constraint is expressed in (7).

$$\forall n \, \varepsilon \, N_c : \sum_{s \, \varepsilon \, S} U_{s,n} \leq U_n \times \mid S \mid \tag{7}$$

### G. Minimizing the Number of Active Gateways - MIN $N_{gw}$

The goal of this optimization is to minimize the number of active gateways in the network. This minimization results in an improved wireless resource efficiency as well as energy and cost savings. Moreover, since gateways could be placed in a sleep state it contributes to an interference reduction. By using the gateway utilization decision variable $U_{gw}$, the minimization can be expressed as shown in (8).

$$min \sum_{gw \, \varepsilon \, N_{gw}} U_{gw} \tag{8}$$

The binary decision variable $U_{gw}$ is subject to additional constraints, ensuring that it only takes on value 0 if there is no end device sending requests for an IoT application through that gateway. This constraint is expressed in (9).

$$\forall gw \, \varepsilon \, N_{gw} : \sum_{ed \, \varepsilon \, N_{ed}} U_{ed,gw} \times R_{ed} \leq \sum_{a \, \varepsilon \, A} D_a \times U_{gw} \tag{9}$$

### H. Minimizing Hop Count Between Computational Nodes and End Devices - MIN $H$

This optimization objective is related to low latency in the communication between computational nodes where communicating services that compose an IoT application are running and the end device that requested the IoT application. This optimization can be achieved by minimizing the hop count between computational nodes and end devices. This minimization can be expressed as shown in (10) by using the Hop Count matrix $H$ and the placement matrix $P$.

$$\begin{aligned} min \sum_{n,ed \, \varepsilon \, N_c, N_{ed}} & H_{n,ed} \\ \times ( \sum_{a \, \varepsilon \, A} \sum_{r \, \varepsilon \, R_a} \sum_{s \, \varepsilon \, S} & P^{a,r}_{s,n} \times \Phi_{a,r,ed}) \end{aligned} \tag{10}$$

### I. Minimizing Path Loss - MIN $PL$

The objective of this optimization is to minimize the path loss of the wireless communication links. The path loss matrix $PL$ is calculated based on the path loss formula for 802.11ah networks, when an urban macro deployment and a central frequency $(f_c)$ of 900 MHz are considered. This formulation can be expressed as in (11). The distance (in meters) is given by the Distance matrix $D$, which indicates the distance between end devices and gateways. This way, by using the path loss matrix $PL$ and the end device execution matrix $U$ the minimization objective is given by (12).

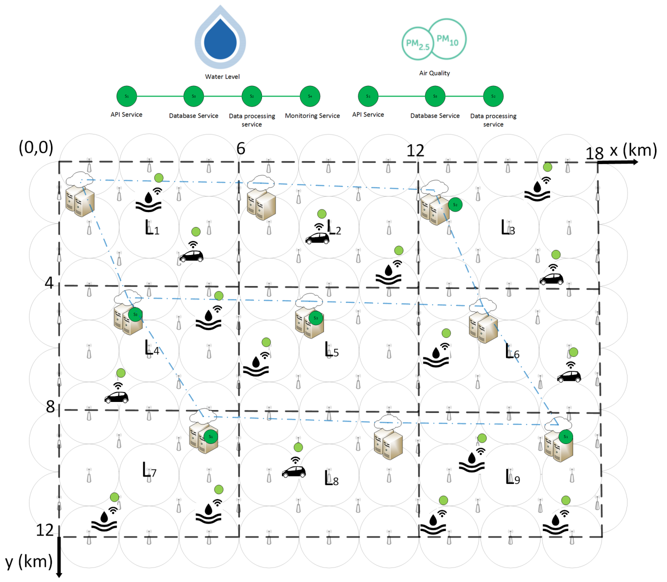

Fig. 2. Evaluation Scenarios.

$$PL(dB) = 8 + 37.6 \log_{10}(d) \tag{11}$$

$$min \sum_{gw,ed \, \varepsilon \, N_{gw}, N_{ed}} PL_{gw,ed} \times U_{gw,ed} \tag{12}$$

## IV. Evaluation Scenarios

The evaluation scenarios are based on use cases within the scope of Antwerp's City of Things testbed [22]. A rectangle area of 216 km$^2$ similar to the area of Antwerp has been considered. Gateways have been strategically placed covering the entire area while minimizing interference due to low coverage overlap between gateways. Two use cases have been evaluated, a static and a dynamic scenario as shown in Figure 2. It should be noted that nine areas of 24 km$^2$ are considered as possible locations for fog resources.

### A. Static Scenario

As a future use case, water level sensors will be installed in sewers in the city of Antwerp. This use case is a static scenario related to a Water Level IoT application, which is decomposed in four communicating services:

- API service responsible for receiving sensor data.
- Database service for storing information.
- Data processing service for information analysis.
- Monitoring service that supervises all other services.

### B. Dynamic Scenario

As an initial proof of concept of the Antwerp's City of Things architecture, air quality sensors have been installed on cars driving around the city of Antwerp [22]. These sensors send measures of typical gasses and climate data such as

TABLE III
INPUT VARIABLES FOR EACH SCENARIO

| Variables | Static | Dynamic |
|---|---|---|
| $N_{ed}$ | 1000 | 100 |
| $A$ | 1 | 1 |
| $S$ | 4 | 3 |
| $N_c$ | 45 | 45 |
| $N_{gw}$ | 123 | 123 |
| $N_f$ | 9 | 9 |

temperature and humidity, which are then annotated with GPS locations. This use case is a dynamic scenario related to an Air Quality IoT application, which is decomposed in three services: an API service, a database service and a data processing service.

### C. Evaluation Setup

The ILP model presented has been implemented in Java using the IBM ILOG CPLEX ILP solver [23]. Input variable values for each scenario are presented in Table III. Nine fog clouds $f \,\varepsilon\, N_f$ each one managing 5 computational nodes $n \,\varepsilon\, N_c$ have been considered. Moreover, 1000 water level sensors randomly distributed in the City of Antwerp have been included in the model for the static scenario while for the dynamic scenario, 100 cars driving around the city have been considered. In the dynamic scenario, each car is driving at an average velocity of 30km/h and each simulation occurs separated by 5 minutes. Therefore, car positions has been changed 2.5 km between simulations. A new $x$ coordinate is randomly selected and then by solving the quadratic equation two solutions for the new $y$ coordinate are obtained. Constraints have been included in the model to make sure that the new calculated car positions are inside the evaluation area.

For both scenarios, the end device $ed$ which makes the $r$th request of an IoT application is randomly selected from the set $N_{ed}$. Every computational node has a CPU and a Memory capacity. The CPU capacity of a computational node $N_c$ represents the processing power per core times the number of cores, which is randomly chosen from the set $\{9, 12, 15, 18, 21\, GHz\}$ while the memory capacity is chosen from the set $\{6, 8, 10, 12, 14\, GB\}$. In the same way, the communicating services have a CPU and a Memory requirement which are chosen from the sets $\{0.5, 0.7, 0.9, 1.1, 1.3\, GHz\}$ and $\{0.5, 0.7, 0.9, 1.1, 1.3, 1.5\, GB\}$, respectively. Each computational node, gateway, end device and fog cloud has a given location $l \,\varepsilon\, L$ associated. A location is one of the nine areas of 24 km$^2$ previously explained in the Section IV. If the location $l$ of a computational node $n$ and a fog cloud $f$ is the same, it means that the computational node $n$ is managed by that fog cloud $f$ since there is only one fog cloud for each location $l \,\varepsilon\, L$. Moreover, $(xy)$ coordinate positions are randomly attributed to each end device $ed \varepsilon N_{ed}$ while for each gateway $gw \,\varepsilon\, N_{gw}$, $(xy)$ coordinate positions are strategically attributed in order to cover the entire evaluation area. Based on these $(xy)$ coordinates, the distance matrix $D$ is calculated by the euclidean distance formula as shown in (13).

TABLE IV
MODEL CONFIGURATIONS - PART I

| A - Wi-Fi Eff. | B - Cloud Energy-Aware | C - Latency |
|---|---|---|
| 1 - MAX $R$ | 1 - MAX $R$ | 1 - MAX $R$ |
| 2 - MAX $SB$ | 2 - MAX $SB$ | 2 - MAX $SB$ |
| 3 - MIN $N_{gw}$ | 3 - MIN $N_c$ | 3 - MIN $H$ |

TABLE V
MODEL CONFIGURATIONS - PART 2

| D - Latency | E - Energy Eff. | DM - Latency | EM - Energy Eff. |
|---|---|---|---|
| 1 - MAX $R$ | 1 - MAX $R$ | 1 - MAX $R$ | 1 - MAX $R$ |
| 2 - MAX $SB$ | 2 - MAX $SB$ | 2 - MAX $SB$ | 2 - MAX $SB$ |
| 3 - MIN $H$ | 3 - MIN $N_c$ | 3 - MIN $H$ | 3 - MIN $N_c$ |
| 4 - MIN $N_c$ | 4 - MIN $N_{gw}$ | 4 - MIN M | 4 - MIN M |
| 5 - MIN $N_{gw}$ | 5 - MIN $H$ | 5 - MIN $N_c$ | 5 - MIN $N_{gw}$ |
| 6 - MIN $PL$ | 6 - MIN $PL$ | 6 - MIN $N_{gw}$ | 6 - MIN $H$ |
| | | 7 - MIN $PL$ | 7 - MIN $PL$ |

$$D(gw, ed) = \sqrt{(x_{gw} - x_{ed})^2 + (y_{gw} - y_{ed})^2} \qquad (13)$$

Then, the path loss matrix $PL$ is calculated based on the path loss formula for the 802.11ah previously shown in (11) by using the calculated distance matrix values. The communication matrix $C$ is a random number between $[0.02, 0.04]$ which represents the bandwidth requirement (in Mbit/s) between two communicating services, $s_1$ and $s_2$, respectively. Moreover, the bandwidth matrix $B$ is a random number between $[6, 14]$ which represents the available bandwidth (in Mbit/s) between two computational nodes, $n_1$ and $n_2$, respectively. The hop count matrix $H$ between computational nodes and end devices is a random number between $[2, 3]$ if the node $n$ and the end device $ed$ are on the same location $l$ or between $[4, 8]$ if the node $n$ and the end device $ed$ are on different locations, $l_1$ and $l_2$, respectively.

Different sets of model configurations for each scenario have been evaluated. In each iteration of the model, a different optimization objective has been considered. In Table IV and Table V, the model configurations are shown. It should be highlighted that for all model configurations first, the number of accepted IoT application requests is maximized and then the satisfaction of the service bandwidth is maximized, ensuring that the communication requirements between application services are guaranteed as well as possible. In Table IV, model configuration $A$ corresponds to a wireless efficiency strategy, since the final objective is the minimization of the number of active gateways on the network. Secondly, model configuration $B$ is related to energy efficiency in the cloud environment, since the final goal of this configuration is the minimization of power consumption of the computational nodes. Finally, model configuration $C$ corresponds to a low latency strategy based on the minimization of the hop count value between computational nodes and end devices. IoT application services are placed closer to the end device when this model configuration is executed.

On the other hand, in Table V, four additional model configuration strategies are shown. Both D and E configurations are composed of six optimization objectives. $D$ prioritizes low latency, while $E$ prioritizes energy efficiency. Moreover, both DM and EM configurations are composed of seven optimization objectives since an additional optimization objective is introduced between the 3rd and the 4th iteration of the configurations D and E, related to the minimization of service migrations on subsequent model iterations in order to reduce delay from reallocating IoT services.

All model configurations have been evaluated 50 times and confidence intervals of 95% have been considered in the evaluation.

## V. Evaluation Results

### A. Static Scenario

The model configurations shown in Table IV have been evaluated for the static scenario presented in Section IV. In Figure 3, the execution speed of the model configurations is shown. By increasing the number of requests, the execution time of the model configurations increases. For 200 requests, each model configuration requires in average at least 23 minutes to find the optimal solution. In Figure 4, the ratio of active gateways and the ratio of active computational nodes for each model configuration are illustrated. Regarding gateways, all are active for configurations $B$ and $C$ since no optimization objective is included regarding wireless efficiency. However, for configuration $A$ whose final objective is related to wireless efficiency, the ratio of active gateways slightly increases with the increase of requests in the network. Results show that 50% of the gateways are active for 140 requests and that only for values above 220 requests, the ratio is higher than 60%. This configuration shows a higher wireless efficiency when compared to the other configurations. On the other hand, the ratio of active computational nodes is 95% independent of the number of requests for configurations $A$ and $C$ due to the lack of an optimization objective regarding energy efficiency in the cloud environment in both configurations. However, for configuration $B$ whose final objective is energy efficiency in the cloud domain, for 20 requests only 9% of the computational nodes are active. Moreover, only for values above 180 requests, at least 80% of the computational nodes are active.

In Figure 5, the average hop count between computational nodes and end devices for each model configuration is shown. Configuration $C$ obtained lower hop count values when compared to configurations $A$ and $B$ due to the fact that the final objective of $C$ is low latency. $C$ achieved slightly constant hop count values of 2.2 while $A$ and $B$ achieved hop count values between 4.0 and 5.5, which results in increased latency in the communication since in average two more hops are required. Moreover, it should be noted that the average hop count decreases while requests increase due to the fact that more computational nodes are needed in these conditions and therefore hop count decreases even if in the model configuration there is no optimization objective related to latency.

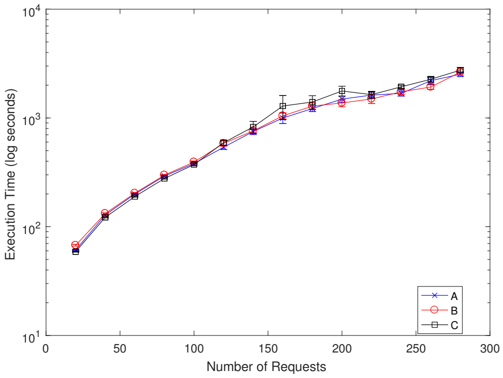

Fig. 3. Execution speed of the model configurations.

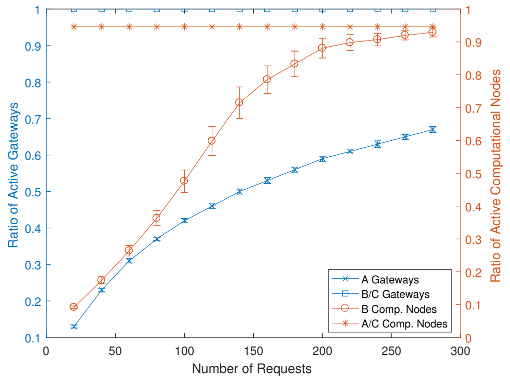

Fig. 4. Gateway & Comp. Node Activity for each model configuration.

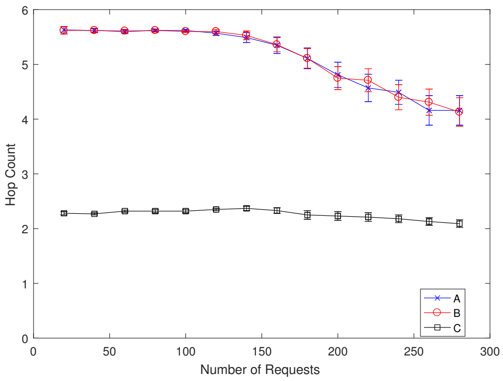

Fig. 5. Average Hop Count value for each model configuration.

### B. Dynamic Scenario

The four model configurations shown in Table V have been evaluated using the dynamic scenario presented in Section IV. In Figure 6, the ratio of active computational nodes for each model configuration is shown. Configurations $E$ and $EM$ achieved the same results. Therefore, minimizing migrations in subsequent iterations of the model, when energy efficiency is more important than low latency, does not alter the final solution. The ratio of active computational nodes is independent of the cars repositioning and remains constant at 16%. However, configurations $D$ and $DM$ obtained different

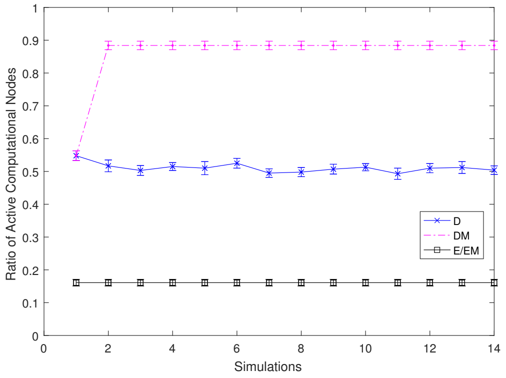

Fig. 6. Comp. Node Activity for each model configuration.

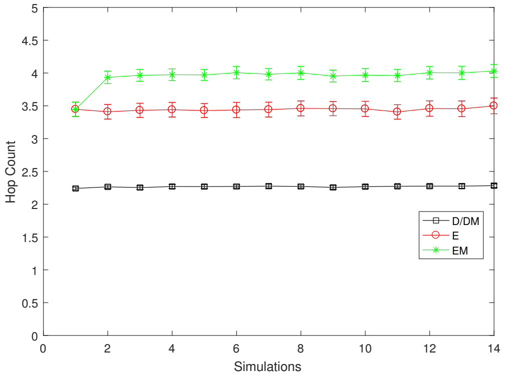

Fig. 7. Average Hop Count value for each model configuration.

results. For configuration $D$, where service migrations are not taken into account average values of 51% are obtained while for configuration $DM$ where service migrations are taken into consideration average values of 88% are achieved. This way, energy consumption cannot be further minimized when reallocations are considered, which will contribute to a higher number of active computational nodes as it was observed.

In Figure 7, the average hop count between computational nodes and end devices for each model configuration is shown. Configurations $D$ and $DM$ obtained the same results because the minimization of hop count occurs earlier than the minimization of service migrations and therefore there is no difference in the achieved results for both configurations. Hop count values of 2.25 are then obtained. However, configurations $E$ and $EM$ achieved different results. Hop count values of 3.5 are obtained for model $E$ while for model $EM$ hop count values of 4.0 are achieved. This is due to the fact that when service reallocations are considered, latency cannot be further minimized, which will contribute to an even higher hop count as it was observed. In Figure 8, the ratio of service migrations for each model configuration is illustrated. Values of 0% for configurations $DM$ and $EM$ are achieved while for $D$ and $E$, average values of 22% are obtained meaning that in order to satisfy the optimal solution, 22% of the communicating services must be reallocated. This way, if both low latency and energy efficiency management strategies are considered, delay caused by service reallocation should be taken into account in the resource provisioning.

## VI. Conclusion

In this paper, an ILP model for the resource provisioning of IoT application services in Smart Cities has been presented. In the last years, the need for resource management strategies for Smart Cities is increasing due to the deployment of IoT application use cases. Proper resource allocation is required in order to minimize costs and maximize QoS. Our model considers not only cloud infrastructure requirements but also characteristics coming from wireless aspects in order to deal with these challenges. The model is executed iteratively since it optimizes multiple objectives, such as latency, service migrations and energy efficiency.

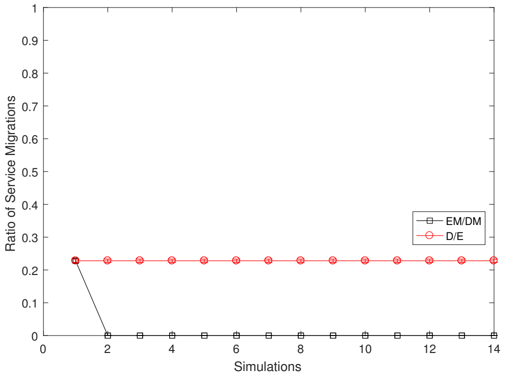

Fig. 8. Service migrations between 3rd and 4th objective.

Obtained results show that there is a clear trade-off between low latency and low energy consumption. For an IoT application service with real-time constraints low latency may be crucial and therefore a low hop count value between the allocated service and the end device must be achieved. However, another IoT application service without real-time constraints could be allocated far from the end device with the goal of minimizing energy consumption since low latency is not an important requirement from that application service.

The result of this work can serve as a benchmark in research related to placement issues of IoT application services in Fog environments since the model approach is generic and applies to a wide range of IoT applications. As future work, the ILP model will be validated through realistic evaluations based on real service deployments.

## Acknowledgment

This research was performed partially within the projects ”Service-oriented management of a virtualised future internet” and ”Intelligent DEnse And Longe range IoT networks (IDEAL-IoT)” under Grant Agreement #S004017N, both from the fund for Scientific Research-Flanders (FWO-V). The computational resources (Stevin Supercomputer Infrastructure) and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by Ghent University, FWO and the Flemish Government – department EWI.

# References

[1] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet of Things journal*, vol. 1, no. 1, pp. 22–32, 2014.

[2] V. Albino, U. Berardi, and R. M. Dangelico, "Smart cities: Definitions, dimensions, performance, and initiatives," *Journal of Urban Technology*, vol. 22, no. 1, pp. 3–21, 2015.

[3] (2017) Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016–2021 White Paper. [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.pdf

[4] M. Yannuzzi, R. Milito, R. Serral-Gracià, D. Montero, and M. Nemirovsky, "Key ingredients in an iot recipe: Fog computing, cloud computing, and more fog computing," in *Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), 2014 IEEE 19th International Workshop on*. IEEE, 2014, pp. 325–329.

[5] L. M. Vaquero and L. Rodero-Merino, "Finding your way in the fog: Towards a comprehensive definition of fog computing," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 5, pp. 27–32, 2014.

[6] H. I. Kobo, A. M. Abu-Mahfouz, and G. P. Hancke, "A survey on software-defined wireless sensor networks: Challenges and design requirements," *IEEE Access*, vol. 5, pp. 1872–1899, 2017.

[7] S. Li, L. Da Xu, and S. Zhao, "The internet of things: a survey," *Information Systems Frontiers*, vol. 17, no. 2, pp. 243–259, 2015.

[8] S. Hachem, A. Pathak, and V. Issarny, "Service-oriented middleware for the mobile internet of things: A scalable solution," in *IEEE GLOBECOM: Global Communications Conference*, 2014.

[9] K. Velasquez, D. P. Abreu, M. Curado, and E. Monteiro, "Service placement for latency reduction in the internet of things," *Annals of Telecommunications*, pp. 1–11, 2016.

[10] M. A. Al Faruque and K. Vatanparvar, "Energy management-as-a-service over fog computing platform," *IEEE Internet of Things Journal*, vol. 3, no. 2, pp. 161–169, 2016.

[11] (2017) SmartSantander FP7-ICT-2009-5-257992 Project. [Online]. Available: http://www.smartsantander.eu/

[12] L. Sánchez, V. Gutiérrez, J. A. Galache, P. Sotres, J. R. Santana, J. Casanueva, and L. Muñoz, "Smartsantander: Experimentation and service provision in the smart city," in *Wireless Personal Multimedia Communications (WPMC), 2013 16th International Symposium on*. IEEE, 2013, pp. 1–6.

[13] D. P. Abreu, K. Velasquez, M. Curado, and E. Monteiro, "A resilient internet of things architecture for smart cities," *Annals of Telecommunications*, pp. 1–12, 2016.

[14] M. Aazam, I. Khan, A. A. Alsaffar, and E.-N. Huh, "Cloud of things: Integrating internet of things and cloud computing and the issues involved," in *Applied Sciences and Technology (IBCAST), 2014 11th International Bhurban Conference on*. IEEE, 2014, pp. 414–419.

[15] S. Rani, R. Talwar, J. Malhotra, S. H. Ahmed, M. Sarkar, and H. Song, "A novel scheme for an energy efficient internet of things based on wireless sensor networks," *Sensors*, vol. 15, no. 11, pp. 28 603–28 626, 2015.

[16] H. Moens, B. Hanssens, B. Dhoedt, and F. De Turck, "Hierarchical network-aware placement of service oriented applications in clouds," in *Network Operations and Management Symposium (NOMS)*. IEEE, 2014, pp. 1–8.

[17] F. Wuhib, R. Yanggratoke, and R. Stadler, "Allocating compute and network resources under management objectives in large-scale clouds," *Journal of Network and Systems Management*, vol. 23, no. 1, pp. 111–136, 2015.

[18] A. D. Mohit Taneja, "Resource aware placement of iot application modules in fog-cloud computing paradigm," in *IFIP/IEEE International Symposium on Integrated Network Management*, 2017.

[19] W. Sun, M. Choi, and S. Choi, "Ieee 802.11 ah: A long range 802.11 wlan at sub 1 ghz," *Journal of ICT Standardization*, vol. 1, no. 1, pp. 83–108, 2013.

[20] S. Aust and T. Ito, "Sub 1ghz wireless lan propagation path loss models for urban smart grid applications," in *Computing, Networking and Communications (ICNC), 2012 International Conference on*. IEEE, 2012, pp. 116–120.

[21] E. Khorov, A. Lyakhov, A. Krotov, and A. Guschin, "A survey on ieee 802.11 ah: An enabling networking technology for smart cities," *Computer Communications*, vol. 58, pp. 53–69, 2015.

[22] S. Latre, P. Leroux, T. Coenen, B. Braem, P. Ballon, and P. Demeester, "City of things: An integrated and multi-technology testbed for iot smart city experiments," in *Smart Cities Conference (ISC2), 2016 IEEE International*. IEEE, 2016, pp. 1–8.

[23] (2017) IBM CPLEX ILOG Optimization Studio 12.7. [Online]. Available: http://www-03.ibm.com/software/products/en/ibmilogcpleoptistud