Received <day> <Month>, <year>; | Revised <day> <Month>, <year>; | Accepted <day> <Month>, <year>
DOI: xxx/xxxx

ARTICLE TYPE

Mobile-aware Dynamic Resource Management for Edge Computing.

Sonja Filiposka[1] | Anastas Mishev[1] | Katja Gilly*[2]

[1]Ss. Cyril and Methodius University, Faculty of Computer Science and Engineering, Skopje, Macedonia
[2]Miguel Hernández University, Dep. of Physics and Computer Architectures, Elche, Alicante, Spain

Correspondence
*Katja Gilly, Miguel Hernández University, Dep. of Physics and Computer Architectures, Avda. de la Universidad, s/n, 03202, Alcudia Building. Elche (Alicante)
Email: katya@umh.es

Abstract

Shifting cloud computing capabilities close to the edge enables provisioning of low latency location-based Internet services that are adapted to user behaviour. However, this can be achieved neither with a simple move of the physical hosts closer to edge networks, nor continuing to abide by the same principles as the ones implemented in traditional cloud computing approaches. In order to accomplish the promised high quality of service, changes must be made to the resource management techniques so that they are adapted to the requirements of fog computing. This paper introduces a novel location-based handoff management and its corresponding implementation of dynamic resource management modules that introduce resource allocation and migration strategies adapted specifically to fog computing. The aim is the implementation of the follow-me behaviour of edge resources considering a tightly coupled perspective with the user location. The proposal is based on the concept of mapping physical areas to logical resource communities while triggering migration of resources to the corresponding logical community when mobile nodes move from the boundaries of one physical area to another. The results analysis from the simulation-based scenarios shows that the effectiveness of our community-based dynamic resource management proposal in following the geographical trajectory of mobile users with wearable devices is over 90% and that it copes very well for highly saturated environments. The comparison with traditional resource management techniques clearly presents the advantages of our proposal, while the parameter wise in-depth analysis discusses its dependencies on the number of mobile nodes, speed, and available resources.

KEYWORDS:
Edge/fog computing, wireless access, dynamic resource management, node mobility, handoff

1 | INTRODUCTION

Fog and edge computing are getting increasing consideration as the online services demand continues growing. Ever since its first mentioning in 2012[?] that introduced the term "simply because the fog is a cloud close to the ground”, the main aim of the fog implementation is to widen the idea of cloud computing to considering, apart from the processing, storage and network capabilities of the cloud datacentre, the location of the service demand and, hence, the network delay. More efforts from academia and industry in providing fast online services are being demanded to satisfy the constant generation and consumption of data from the many and various mobile devices that compound the Internet of Things (IoT)

In order to reduce real time responses and provide location-aware services, it is required to add some additional layers of computational and network resources closer to the edge of the network[?]. As an essential technology, Multi-access Edge Computing (MEC) is considered a broadband network improvement in terms of efficiency that promotes the 5G evolution[? ?].

Many recent research works have been directed towards the definition of a general fog reference architecture, such as[?]. The objective is to obtain a more efficient operation by defining a deployment type based on an n-tier hierarchy that would enable diverse localisation levels and resource distribution, that ranges from the edge servers that are locally deployed with each base station or wireless access point (AP) for minimum latency responses as the lowest level, through grouping the edge datacentres into neighbourhoods, to finally connecting to the remote cloud that provides computing and storage resources for long term planning, business analytics, etc.

Traditional cloud resource management problems have been widely researched during the last decade and some of the solutions proposed for obtaining optimal usage of the virtualised pool of computing resources can be applied to manage a multitude of fog computing nodes dispersed in the communication provider network. The physical fog nodes will host virtual machines (VMs) or virtual containers that will be in charge of providing the service response on real time to mobile user requests. Fog specific resource management problems include particular attributes as the user location and its mobility. In order to ensure the minimum latency, the decision about where to place the virtualised resources is based on the current wireless connection to the network APs. This makes the fog management particularly dependent on the geographical location of the IoT nodes. The main problem in case of mobile devices is the implementation of "follow me” services that can be supported by live VM or container migration through the available fog nodes so that computing resources are always as close to the mobile device as possible[?].

While migration of live services is getting more interest in the research literature[?], the definition and analysis of efficient virtual resource placement and migration algorithms that will ensure the maximum performance in the sense of optimal resource usage and the minimum latency, is still in its formative stages. Towards this goal, we deal in this paper with the problem of fog resource management by taking advantage of the fog hierarchical architecture and considering the mobile device localisation in both geographical and virtual logical communities with different levels of granularity. The results analysis shows that the proposed approach enables highly optimised solutions for both virtual resources placement and migration problems that are dynamically adapted to the end user pattern of movement continuously providing high Quality of Service (QoS).

The rest of the paper is divided as follows: In section 2, the fog computing platform and its components are introduced, after which the fog resource management problem is discussed and the recent related work in this field is presented. Section 3 proposes a community-based fog architecture for efficient dynamic resource management that combines the information from the modules for quality of service monitoring and mobility tracking and feeds it to the VM management module. Section 4 discusses the implementation and presents the results from different simulation scenarios that compare the proposed fog resource management approach to other traditional techniques and provide insight into the effect of node density and speed on the efficiency of the solution. Finally, Section V concludes the paper providing directions for future work.

2 | RESOURCE MANAGEMENT IMPLEMENTATION IN THE FOG PLATFORM

Adding computing and networking layers close to the wireless service demanding devices requires the expansion of provider's communication network through the wireless edge APs. The idea was first introduced under the term of micro datacentres by Bahl in[?], where the traditional AP cells are extended with a small set of nodes, known as edge servers, that provide the ability for local storage and processing. The basic idea is presented on Fig. 1 . These distributed nodes would belong to a virtual overlay network (that can be dynamically managed by the use of technologies such as Software Defined Networks (SDN) and Network Function Virtualisation (NFV))[?]. The importance of implementing these technologies and their impact on the fog architecture is described in other related work on the topic such as[?]. In essence, NFV and SDN enable the implementation of key technical challenges in the fog architecture, such as dynamic allocation, orchestration and migration of groups of virtualised resources across a wide area of interconnected edge networks.

The creation of a fog layer requires an implementation of a complete fog computing platform that will provide the benefits of the introduction of the close-to-the-edge computing resources. Each fog node may require additional computing power in the fog infrastructure, represented as application services placed in individual VMs on the top of the platform (see Fig. 2). These VMs reside on the available edge servers that represent the available fog infrastructure in the edge network. The edge servers are based on the virtualisation technology, providing virtual resources in the edge network, that are managed by the communication operator in a similar fashion to cloud datacentres, but on a smaller micro scale. The main additional fog related components are represented with the middle layer components on Fig. 2 . The combination of these components implements the required fog computing functionality.

The *location and mobility tracking module* is in charge of tracking the current location of each mobile fog node. This module is responsible of the handoff process in the cases when the end user changes the AP used for the active wireless connection.

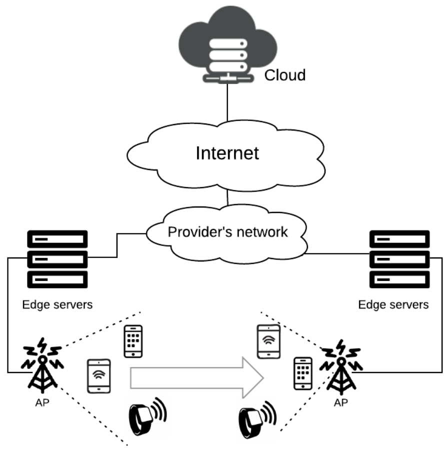


FIGURE 1 Fog architecture with mobile nodes.

Each handoff triggers events that need to be processed by the *QoS monitoring module*. The task of the QoS monitoring module is to continuously check the provided QoS to the end user making sure that there are enough virtual resources available for the requested service and that it is placed as close to the user as possible. As the location and mobility tracking module sends information about a handoff in progress, the QoS module must decide if the handoff has lowered the provided QoS to the user due to increased latency, in which case the VM that hosts the user fog service needs to be migrated to a different cluster of edge servers. In the case that a migration is required, this module will send a trigger event to the *VM management module* that is tasked with the centralised management of the virtualised resources provided by the edge servers. This module implements the resource management techniques used for making the decision of initial placement of requested fog service and migration of the existing VMs. Its goal is to decide the most suitable edge server for hosting a given VM. In order to make the most informed decision it uses the current resources utilisation and the latency measurements provided by the QoS monitoring module. In addition, the *cost analysis module* also uses the information about the utilisation of the resources provided to the end users in order to provide the necessary statistics for billing. This module has an active decision role in the case when resources in the cloud must be used either because there are not enough resources in the fog, or because some type of batch processing such as machine learning training on historical data is required. It is then tasked to select the most cost effective cloud service provider. The *authentication and authorisation module* provides all functionalities related to identifying users and services using different types of credentials, and deciding on the authorisation for use of different elements of the architecture as defined in the active policies. In addition to the security features, this module also sends triggers to the cost analysis module for accounting purposes. Finally, the *communication service module* is responsible for direct exchange of messages with the fog nodes. It receives their resource demands and attends to events in the fog platform related to resource availability and readiness or authentication requirements.

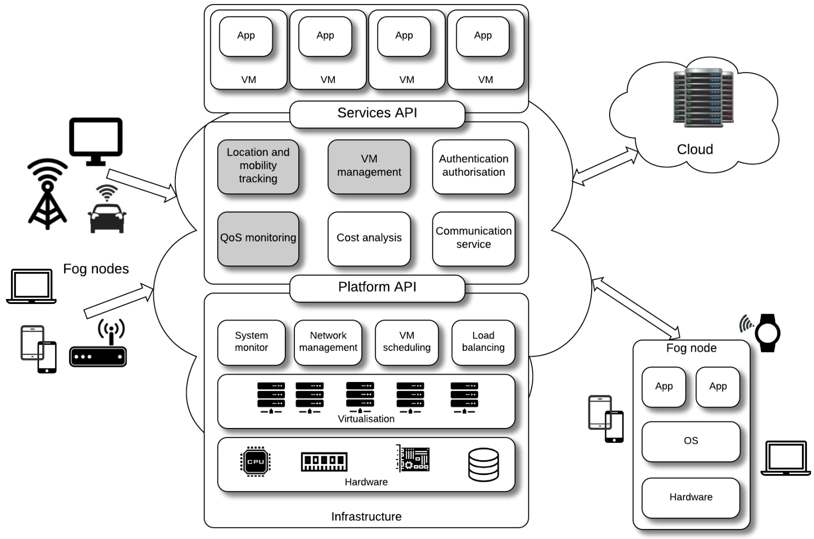


FIGURE 2 Fog computing platform components.

According to the described fog computing platform (see Fig. 2), the resource provisioning in the fog is proposed to be based on high-level orchestration components that ensure the efficient placement of virtual nodes and the dynamic allocation of services, supported by a monitoring system that collects and reports on the behaviour of the resources. The problem of placing a VM in a fog infrastructure composed of distributed clusters of edge servers needs to be solved in two hierarchical steps. The first optimisation in the decision making process must be based on the minimum latency requirement, i.e. to find the required resources at the closest cluster of available edge servers, where the most optimal solution would be the cluster that is directly connected to the current AP used by the mobile node that requested the VM. The second level of optimisation is used for the selection of one edge server from the chosen cluster depending on the network provider's goal, that can be resource load balancing or used resources' consolidation in order to save power.

With the inherent node mobility due to wireless access, the problem of efficient migration must be solved as well. In the fog approach, the migration of virtualised resources is triggered as a response to nodes' mobility in order to achieve minimum latencies, unlike migration processes in the cloud, that are done by the cloud provider in order to consolidate resources and, hence, save power. In other words, whenever a handoff event occurs because the mobile node switches to a different wireless AP, it must be investigated whether a migration of the fog computing resources used by the mobile node to a different fog cluster is required so that high QoS is guaranteed. Obviously the aim is to have the virtual resources on the fog cluster that is as close to the mobile node's new location as possible. For this decision process it is needed to stream the information about node mobility from the *location and mobility module* to the *QoS monitoring module* and then to the *VM management module*. The requirements for migration events, i.e. VM handoff, are analysed in[?] , where a similar layered architecture is proposed based on the location and mobility and VM management components that work at unison to provide efficient resource management.

2.1 | Related work

There is a high demand of resource management algorithm frameworks and implementations that will efficiently handle user mobility and provide optimal usage of the available fog resources. One example of these approaches can be found in[?] where the authors employ a so called cloud atomisation to create highly granular pool of virtual resources. The aim in the described resource placement problem is to provide load balancing on the global level of all fog resources. To achieve this the authors use graph repartitioning of the graph obtained after the cloud atomisation phase. Another suggested approach that is somewhat similar to the one proposed in this paper is provided in[?] where the problem of service allocation in combined fog-cloud n-tier scenarios is investigated with the aim to minimise the latency when accessing services. However, unlike our proposal, in[?] migrations are not considered which renders the initial placement ineffective over time, as it is presented in the results section later on.

Similarly, authors in[?] analyse the fog architecture and its components, wherein they recognise that one of the key modules of the architecture is the resource management module. They focus on the requirements for this component so that the architecture can provide minimum latency and maximum throughput, as these are the most critical features in fog computing deployment scenarios such as health monitoring or emergency response team and large scale IoT solutions. The fog architecture proposed by the authors contains a resource management component which is consisting of a provisioning and scheduling module. The aim of the component is to coherently manage the fog resources so that all QoS requirements defined by the application level are met, while the resource wastage produced during the placement procedure is minimised. In the results section they compare the efficiency of the cloud-only and the edge-ward placement approaches. In[?] the architecture and hierarchical resource management for a cooperative fog computing solution designed to deal with Big Data in the Internet of Vehicles are proposed. While the authors discuss an architecture very similar to the one described here, their emphasise is on the cooperative work of the clusters of edge servers when dealing with mobility and service migration. They propose a hierarchical resource management framework trying to provide energy efficiency and QoS. They discern between intra-edge cluster resource management and inter-edge cluster resource management and they mathematically model the optimal assigning probability matrix based on an M/M/1 queuing model and evaluate the input/output buffer sizes and transmit rates.

In[?], the authors are focusing on IoT scenarios and users. They differentiate between several service types and models for pricing and propose a model for reservation based on estimation of the usage of resources. For these purposes they use the fluctuating relinquish probability of the customer, service type and price. The costs and pricing are discussed for simulation scenarios created in the CloudSim simulator. In[?] a case study that focuses on video distribution and video streaming is presented. The authors propose a resource allocation approach for fog computing that also incorporates the goal for minimising the carbon footprint. The described distributed algorithm optimises the available bandwidth in the network across all user requests for video streaming. Similarly, in[?] the authors discuss a cost efficient resource management in fog computing supported medical cyber-physical systems. They use mixed integer linear programming, transformed into a two phase heuristic algorithm that has an objective to minimise the overall unit cost for deploying in a given base station edge infrastructure so that a QoS is guaranteed to the end user. The results compared to greedy algorithms show significant improvement in optimisation, however requiring significant time to obtain an optimal solution for a larger topology with more than a hundred users.

Compared to the discussed related papers that analyse different aspects for fog resource management, in this paper we focus on the operation and orchestration of three modules that are part of the fog architecture: the location and mobility management module, the QoS monitoring module and the VM management module. Orchestrated together, the three modules are the pillars of dynamic resource management. By responding in concert to handoff events, these modules can ensure the lowest possible latency when accessing the fog virtual resources assigned to the mobile user. This behaviour can be described as a follow-me principle, which is provided using resource placement and management strategies that involve the process of match-making physical service areas provided by the APs to a hierarchy of logical communities built by combining clusters of edge servers.

3 | FOG RESOURCE MANAGEMENT STRATEGIES

Keeping in mind that the details of the general fog architecture are still in the early stages, we present a fog resource management component that is both location-aware, enabling effective initial VM placement based on the requesting user location, and dynamic, activating migration of the VM based on the user mobility making sure that the optimal latency is continuously provided. The proposed solution is based on multiple hierarchical objectives, where lowest latency is the top objective. Thus, effective initial placement in this case can be described as choosing the edge server for hosting the VM so that minimum latency is required for the requesting fog mobile node to access the virtual resources at its disposal. Our solution completely decouples the algorithms used, on one hand, for selecting the optimal edge servers cluster from, on the other hand, the selection of a specific edge server. It, thus, enables the introduction of a second level objective that provides additional possibility for optimisation. This lower layer objective provides means to employ cost saving techniques such as energy efficiency or balanced use of resources that are usually of interest to the fog infrastructure provider. It is important to remark that the proposed strategies for resource management do not depend on the type of virtualisation employed (VMs or containers). However, the results presented below correspond to VMs as containers migration is still proving to be very difficult[?].

The proposal presented in this paper stems from an already tested framework for cloud resource management that uses the notion of virtual and physical hierarchical communities[?]. The original cloud-based framework is based on a two-level hierarchical approach supported by the datacentre network topology and the interconnectivity between VMs in the user requested cloud services. The complete cloud datacentre is divided into a number of overlapping communities of physical hosts by employing the modularity principle? on the network topology through a hierarchical dendrogram. The upper level placement algorithm seeks to place one full cloud service (a set of tightly coupled VMs) to the smallest community with the available required resources. Once the community is chosen, the lower level placement algorithm does the actual VM-to-physical-server placement based on the cloud provider's optimisation goal.

We sum up the virtual to physical community matching procedure done by firstly defining a set of variables in Table 1 , that is used in the two level multidimentional optimisation method proposed by the framework? .

PM	physical machine
r_i	available capacity of the resource i
cr_i	total capacity of the resource i
d	number of resources
VM	virtual machine
s_j	requirement for the resource j
Com_i	physical community i
c	number of communities
n_i	number of PMs in Com_i
CS	virtual community
cs	number of VMs in a CS
D	matrix representation of the hierarchical dendrogram

TABLE 1 Definition of the variables used

On one hand, a physical machine is defined as $PM = (r_1, r_2, ..., r_d)$ and the total capacity of a PM is $(cr_1, cr_2, ..., cr_d)$, being $0 \leq r_i \leq cr_i$. A VM is defined as $VM = (s_1, s_2, ..., s_d)$, being $0 < s_i \leq cr_i$. A physical community is defined as $Com_i = \{PM_j | j \in [1, n_i]\}$, for $i \in [1, c]$ and the boolean variable $a_{i,j}$ indicates if PM_j belongs to Com_i, providing a matrix A which has the graphical form of a dendrogram. On the other hand, a virtual community is defined as $CS = \left\{VM_k | k \in [1, cs]\right\}$.

The objective is to define the best possible match as the smallest physical community that can accommodate the complete virtual community under all fixed constraints, so its defined an aggregate function $\mathcal{G}_{F(a_{ij}PM_j)}(Com_i)$, and its equivalent for virtual communities, where $F(x)$ is the specific aggregation function.

$$PM_i^{'} = \mathcal{G}_{F(a_{ij}PM_j)}(Com_i) = \sum_{j=1}^{d} a_{ij} PM_j$$
$$VM^{'} = \mathcal{G}_{F(VM)}(CS)$$

The high level optimisation consists on finding the community $PM_i^{'}$ that optimises the higher level placement function $F_{HL}(PM_i^{'}, VM^{'})$ and then, the second optimisation level is defined for setting the placement of individual VMs from the virtual community into the PMs of the previously selected community by this lower level placement function $F_{LL}(Com_i, CS)$.

The cloud-based framework highly efficiently improves the network and energy performance while consolidating cloud servers (more details on the full implementation can be found in[?]).

In this paper, we adapt the concepts of physical and logical communities matching and a two level optimisation to fog computing with the high-level aim of minimising the VM based service-fog node latency, and thus, providing a high quality of experience (QoE) to the end-user.

We represent the network infrastructure of the communication provider that supports fog services by using the previously mentioned hierarchical dendrogram of overlapping communities of physical servers. The lowest atomic layer represents the smallest communities that correspond to one cluster of edge servers directly connected to one or multiple co-located wireless APs covering one service area while the root of the dendrogram is the community of all the nodes in the network. Basically, on the lowest layer, the edge servers and their corresponding serving wireless access points are connected to one network switch. These lowest layer logical communities of edge servers are then interconnected using higher layers of distribution and root level switches. On the higher levels, based on the provider's physical (or preferably SDN based virtual) network topology, the atomic logical communities are combined into larger ones that correspondingly cover wider geographical areas. The root of the dendrogram is the whole network with all of its available edge servers represented as one logical community. In order to obtain the dendrogram, it can be used a community detection algorithm such as[?] using as an input the graph representing the communication provider network with all links and nodes. Each logical community has a well defined pool of available and used virtualised resources that reside on a number of edge servers.

To implement the adaptation of the concept, we represent the hierarchical dendrogram in the form of a matrix D defined as

$$D = [d(i,j)]; i \in \{1 : N_{AP}\}; j \in \{1 : H\};$$

where N_{AP} represents the number of lowest layer logical communities of edge server directly connected to at least one AP and H is the height of the dendrogram, where on level H is the root of the dendrogram with one community encompassing all nodes. Each element $d(i,j)$ in D is defined to represent the community id Com_i, using the following definition

$$d(i,j) = \begin{cases} Com_j & \text{if } i = 1 \\ Com_k & \text{if } i > 1; Com_i >_V Com_k \end{cases}$$

where the $>_V$ operator represents the "is child of" relation in the dendrogram.

In the case of fog computing, the main difference in the resource management framework is reflected in the rules for the higher level optimisation. While the cloud oriented framework is mainly focused on overall optimal resource utilisation in the datacenter disregarding location, fog computing requires achieving the lowest possible latency between the user and the VM which is location dependent. This change in the high level optimisation is reflected by considering only a subset of the full dendrogram, i.e. the hierarchical list of communities that contain the AP currently being in use by the user that requested the fog service. This subset can be represented as the row vector, $D(AP, :)$, where AP is the id of the lowest level community that encompasses AP.

Using the new row vector $D(AP, :)$, we can transform the previous equation to

$$PM_i^{'} = \mathcal{G}_{F(a_{ij}PM_j)}(Com_i \in D(AP, :)) = \sum_{j=1}^{d} a_{ij}PM_j$$
$$VM^{'} = \mathcal{G}_{F(VM)}(CS)$$

In this way we effectively narrow the search of Com_i to the defined subset $D(AP, :)$ in order to identify the smallest community containing AP that will fit the requested fog service CS.

In order to visually depict the problem, the dendrogram can then be projected over the geographical service area based on the coverage area of each wireless AP. Part of this process is depicted in Fig. 3 , where only the lowest level communities are presented. The coverage area of each wireless AP is represented using a Voronoi diagram? that effectively assigns each coordinate in the service area to the closest AP based on the wireless signal strength. If multiple APs are connected to the same cluster of edge servers, then the corresponding service area that is related to this community is defined as the geographical area that represents a union of the service areas of all individual APs that belong to the community. Based on this definition, the geographical area that is served by the fog provider can be described as a union of service areas, each mapped to a separate logical community. If the lowest layer of the dendrogram is projected, as in Fig. 3 , then the union consists of non-overlapping service areas, where the corresponding logical communities are composed of separate clusters of edge servers.

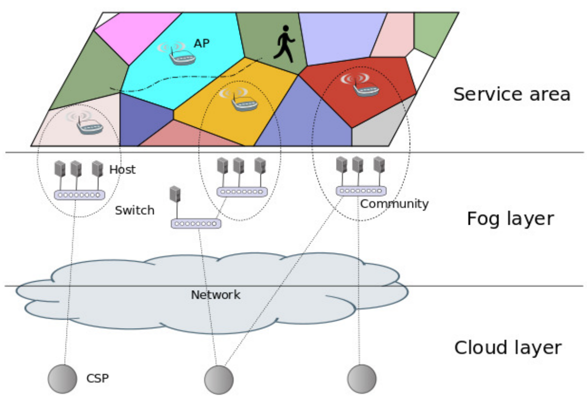


FIGURE 3 Fog nodes communities on the Voroni-geographical area

As discussed previously, the main goal of the resource management module is to provide the lowest possible latency between the VM and the requesting fog node. This is done by using the mapping between physical service areas and lowest layer logical communities, that are connected in a 1:1 relationship. The higher layers enable providing as optimum as possible placements in the cases when there are no resource available in the most optimal logical community. However, in order to continue providing low latency, the optimality of mapping between the physical service area used by the fog node and the logical community that hosts the VM must be maintained using live VM migration techniques if necessary. The migration process moves the VM from one edge cluster to another, which can be relatively close as measured by their distance in the dendrogram, of several hops apart. The physical area to logical community mapping is such that physical closeness does not impose logical closeness, in which case the migration will move the VM in a different branch of the dendrogram tree. In other words, the fog network infrastructure is typically created using a tree-like pattern where the core (root) switching layer that sits on top is divided in a number of distribution layer service areas that are separated again on the lowest edge layer covering non-overlapping geographical service areas with separate APs. Therefore, each handoff event representing a change of service area, may trigger a migration request for changing the current logical community that hosts the fog node VM to a new one potentially x hops away.

3.1 | Allocation or initial placement of fog resources

VM allocation can be considered a subset of optimal resource usage problems[?]. There have been many efforts in developing heuristics trying to optimise a given cost function, even when very high computational complexity is involved. A cost objective function represents the heuristics optimisation goal while considering a given set of requirements represented as constraints. The constraints are used to describe the suitability of a given resource to be considered as an available resource when comparing it to the requested resources necessary to instantiate a VM. Older heuristic algorithms in use could only be used for resources that are represented with a single parameter, either the most important one or defined as a simple function of a multitude of parameters. This approach effectively analysed the solutions space using a single optimisation goal. Recently, however, an extension based on vector spaces has enabled the description of more complex optimisation problems. using this approach, the computational resources of edge servers and VMs can be described using multiple parameters, each representing a different dimension in the vector space. In this way resource attributes such as OS, storage, CPU, RAM, etc. are described, and the optimisation goal can be defined as a multi-objective function.

Considering the information provided by the dendrogram that includes the hierarchical community structure, the decision about the optimal VM placement against the overall pool of available resources (i.e. based on the minimised *mobile node - serving VM latency*) can be solved in a two-step process:

(1) find the smallest community that contains the currently used wireless AP and that has enough available resources to allocate the user request,

(2) use load balancing, server consolidation, or any other traditional VM placement algorithm that will decide where to place the requested VM(s) given the available edge servers that belong to the chosen community.

Hence, placement decision can be described as a multi-objective function that is restrained to the hierarchical sets of available virtualised resource pools. It can be identified as a packing problem and solved employing a technique based on vector algebra, while using as many dimensions as needed to describe all physical and virtual resources characteristics.

The optimal solution would be that the chosen community is on the lowest hierarchical layer and hence, it is as close to the mobile node as it is possible with the given network topology. Worst case scenario where placement is still in the fog layer would mean that the placement algorithm is applied at the root of the dendrogram with the whole fog network seen as one community effectively employing only the second level optimisation. The ultimate worst case scenario is when there is not available resources in the fog, and the request has to be forwarded to the cloud.

The basic workings of the initial placement algorithm are presented in pseudocode Algorithm 1, where the function *chooseEdgeServer* represents the call to the chosen second level VM placement algorithm.

```
int VMPlacement(VM vm, AP currentAP, Communities[] comDendrogram){
    int locationPlaced = -1; // no location found
    Communities[] potentialCom = findComWithAP(comDendrogram, currentAP);
    if (potentialCom.size() != 0) {
        int index=0;
        while (index <= potentialCom.size()) {
            if (resourcesAreAvailable(vm, potentialCom(index)))
                locationPlaced = chooseEdgeServer(vm, potentialCom(index));
            else // no match, go up the hierarchy
                index = index + 1;
        }
    }
    if (isEmpty(locationPlaced))
        locationPlaced = placeInCloud(VM);
    return locationPlaced;
}
```

Listing 1: Initial VM placement pseudocode

3.2 | Migration of fog resources

In the cases when the node that requests the VM is mobile and moves throughout the service coverage area, a live VM migration procedure needs to be invoked every time a handoff to a new wireless AP that belongs to a different community occurs.

In this case, the mobile node leaves one coverage area and moves to another one that is optimally served by a different set of edge servers. The corresponding handoff event is triggered in the location and mobility tracking module, and is then sent off to the QoS monitoring module for further analysis. If the recorded handoff can be mapped to a new physical area, i.e. a change of the current mapping to a different logical community, then the trigger is forwarded to the VM management module where it is transformed into a migration request.

The migration process aims at providing the best QoS by trying to migrate the serving VM(s) to the new community that serves the area where the mobile node is now located. If no resources are available in the destination community, the migration process is cancelled. In other words, unlike the case with placement, the algorithm does not explore the space of the upper hierarchical communities (see Algorithm 2). In this way, we avoid unnecessary use of additional resources for migration when the result may be either equal or worse latency compared to not migrating at all. The efficiency of this approach is analysed in the results presented in the next section. The choice for the optimal edge server within the destination community is implemented using a second-level objective algorithm that may be identical to or different from the one used in step (2) during the initial placement process.

There is only one exception to the rules for migration: in the case when a migration due to handoff is requested, and the corresponging VM(s) is hosted in the cloud (there were no available resources in the fog during the process of initial placement or previous migration attempts), the migration process continues along the hierarchical tree of upper layer communities. Starting from the destination AP's community, it will try to find available resources so as to migrate the VM(s) from the cloud to the fog. In this scenario, of course, any placement in the fog will be more efficient compared to not using the fog at all.

The migration techniques used for the implementation of the migration process must be based on live migration so that the mobile node does not loose connectivity to the fog while migration is in progress. However, the sacrifice to be made while enabling continuous access to the VM resources, is the usage of extra resources. The process of live migration effectively uses up virtual resources on both the source edge server that hosts the migrating VM and destination edge server that will host the migrating VM once the process of migration is finished. Only upon completing the migration process are the resources used by the migrating VM released on the source edge server. Therefore, it is imperative that the quantity of fog resources is such that allows for this double resource usage during migration events.

```
int VMMigration (VM vm, AP newAP, Communities[] comDendrogram) {
    int newLocation = -1; // no location found, old one will be used
    if (isInCloud(vm))
        newLocation = initialPlacement(vm, newAP, comDendrogram)
    else { // try to migrate in fog
        Communities[] potentialCom =
                        findSmallestCommunityWithAP(comDendrogram, newAP);
        if (resourcesAreAvailable(vm, potentialCom))
            newLocation = chooseEdgeServer(vm, potentialCom);
    }
    return newLocation;
}
```

Listing 2: VM migration pseudocode

4 | IMPLEMENTATION OF THE PROPOSAL AND SIMULATION RESULTS

In this section we present the results obtained by applying the proposed changes in the popular CloudSim simulator[?] in order to add an extension with the fog architecture modules. The physical and virtual communities concept and the community detection algorithm have also been implemented in the simulator, together with the fog datacentre that is an additional branch of the complex network that was previously developed in CloudSim[?].

A multidimensional approach is used for representing computing nodes considering the number of cores, size of RAM and bandwidth, while networking nodes are implemented as switches. We have configured a higher value of the bandwidth of the links that connect the upper layers (core-distribution layer) at 10 Gb, leaving the lower layers of the network (distribution-edge layer) at 1 Gb, similar to the fat tree topology that is a popular proposal used in distributed computing centres and can be easily configured as an overlay virtual network of the communication provider. There is at least one wireless AP that is directly connected to every edge switch. The core layer switches (top of the dendrogram) are connected to the cloud using the Internet provider connection. In this way, the connection to the cloud is as far away as possible from the mobile fog nodes representing a realistic scenario. The topology of the fog provider network architecture defined as a virtual overlay is specified during simulation setup enabling flexibility for future cross comparison analysis.

We have extended the original set of policies for VM placement and migration implemented in CloudSim to include the fog VM placement and fog VM migration algorithms. These two policies constitute the VM management module. After the simulation starts, as each fog node requests a fog service for the first time, the fog VM initial placement algorithm is invoked. After the initial setup, the calls for resource management are forwarded to the VM migration policy that is invoked with each external handoff event.

The location and mobility tracking module is implemented in such a way that it is aware of the size of the overall service area covered by the fog operator, the total number of mobile nodes/users and the geographical positions of all wireless APs. The input needed for evaluating handoff events for this module can be defined as a real life mobility trace script, or as the output from a mobility generator. The output from the module are the triggered handoff events when mobile nodes switch from one AP to another during their movement throughout the service area.

The generated information of handoff events is sent to the QoS monitoring module. This module checks if the handoff event should be accompanied with a corresponding migration request. In other words, the module determines the new latency between the mobile node and its VM and, in case this value is not optimal, it triggers a migration event that is sent to the VM management module so that the appropriate VM migration policy is invoked.

4.1 | Description of the simulated scenario

We have simulated a number of scenarios that were defined in the fog computing extension of the CloudSim simulator as described in the previous section. The results presented in this section are extracted from the output traces and averaged over several runs.

We consider that each fog service is implemented using a single VM, and every mobile node uses one fog service. The VM size is chosen randomly from three available options for number of cores and RAM size: (1, 1 GB), (1, 2 GB), and (2, 2 GB). All options have a network bandwidth of 100 Mb. These characteristics where chosen to closely follow the fog related scenarios discussed in the related work section. Fog computing layers are represented using a 3 layer fat tree hierarchy that include 10 edge servers per edge switch, and 2 wireless APs are also connected per edge switch. Therefore, the fog network has 8 edge switches in total that compose the 8 lowest level communities (that represent the leaves of the dendrogram). Their interconnections make up 15 communities with 80 hosts in total, where the highest hierarchical community represents the full fog layer (root of the dendrogram). The higher level interconnection between the edge switches is done using 8 additional distribution and 2 core level switches, as it can be seen in Fig. 4 .

As we have chosen a scenario setup that considers the provider's virtual overlay network follows a tree design, we can set three labels for the possible distances between a mobile node and its corresponding serving VM (or VMs in the case we consider compound services) in fog layers can be:

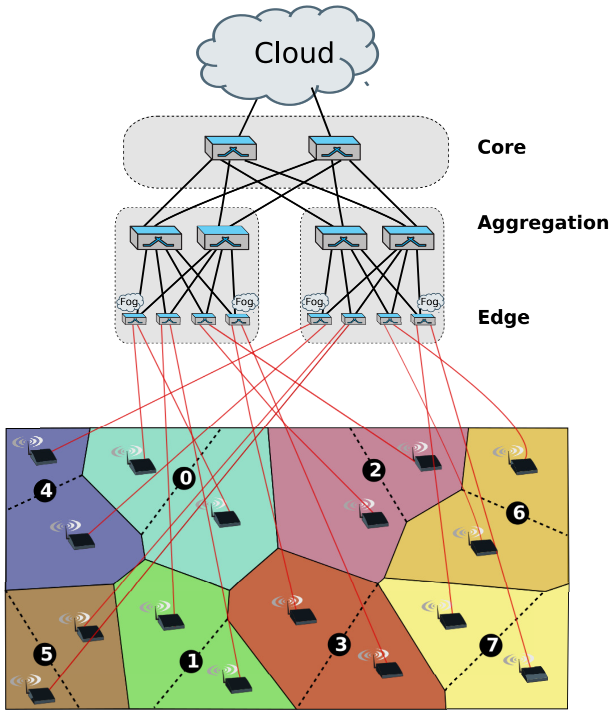


FIGURE 4 Fog network infrastructure layout.

- *optimal*: when the mobile node accesses through the wireless AP that is connected to the same edge switch, this means it is in the same community with the serving VM.
- *1st level hierarchy*: in this case, the mobile node connects to the VM(s) via edge switch, distribution switch and then back to another edge switch, since the service is hosted in a neighbouring community.
- *2nd level hierarchy*: the connection is done via edge switch, distribution switch, core switch, distribution switch, edge switch, since the service is hosted in a more remote community.
- *3rd level hierarchy*: the connection is done via edge switch, distribution switch, core switch, another core switch, distribution switch, edge switch, since the service is in a completely different part of the tree.

We consider the switching delays provided in the network constants for CloudSim to represented the results as they are standard delays for edge, distribution and core layer optical links (see Table 2).

router level	*delay*
edge	1.57 ms
aggregation	2.45 ms
root	2.85 ms

TABLE 2 Delay (in ms) of interconnection devices

Edge servers are configured with 6 cores, 12 GB RAM and 1 Gb bandwidth (one of the predefined server types in CloudSim). We have implemented two different algorithms for the second level optimisation process in the placement decision, that are vector-based load balancing or vector-based server consolidation[?] . We have chosen to present the results for the first case (load balancing option) based on the similarity of results obtained with both options.

The total coverage area size is 800 m x 400 m, and is divided into 8 (4 x 2) equally sized cells. Inside each of the cells, there are 2 wireless APs with randomly selected locations. A Voronoi diagram shows the final coverage of each AP (see Fig. 4). The coverage range for each AP is set to be 200 m. Each of the 16 AP coverage areas are mapped to one of the lowest level communities. Two possibilities of mapping are used in the provided simulation results, in order to analyse whether the mapping function has any effect on the obtained performances. These two ways of mapping physical area to logical community are presented in Fig. 5 , wherein the location of each AP is represented as a dot, while the mapped logical community is represented using different colours and numbers. The two chosen mapping functions are such that the centre of the coverage area belongs to distant parts of the dendrogram, i.e. the overlay fog provider network, for mapping 1, and to the same part of the dendrogram for mapping 2.

(a) Mapping 1 **(b)** Mapping 2

FIGURE 5 Mapping physical coverage area served by two co-located APs to a logical fog community served by 10 edge servers

All mobile nodes in the simulations are moving according to the Random Waypoint mobility model. The total number of nodes varies from 160 to 320, and the average speed of the nodes can be 1 or 2 m/s (corresponding to leisure and fast paced walks).

4.2 | Results and discussion

We have obtained five sets of simulation traces by the mobility module of 160, 200, 240, 280 and 320 mobile nodes and introduced them in Cloudsim obtaining around 50% usage of available resources for 160 nodes, 75% for 240, and 100% for 320 mobile nodes (with less than 1% VM hosted in the cloud). These scenarios have been chosen in order to stress the system and observe the success rate of migrations in cases when there are diminishing available resources across the edge servers when implementing the follow-me behaviour.

Several types of analysis have been conducted on the obtained traces: number of successful and failed migration attempts over time and its effect on the delay, follow-me efficiency cross-comparison with traditional approaches used for placement and/or migration, the use of different mappings when defining the relationship between physical areas and logical communities, influence of the node speed on the performance and energy efficiency analysis of the fog computing infrastructure. All of the following results are based on scenarios that use Mappin 1 unless stated otherwise. All presented results are configured to use second-level placement and migration algorithm based on load balancing that aims to balance the load on the edge servers in use.

4.3 | Follow-me efficiency

Initially, we analyse the performance of the initial placement of the requested fog services together with the migration request events that are triggered every time the mobile node handoff process results with a connection to an AP that is served by a different edge servers community. In Fig. 6 , the efficiency of the placement and migration processes is presented via the number of optimally placed and migrated VMs and the obtained average delay for the example case of 240 mobile nodes moving with an average speed of 1 m/s and 2 m/s.

(a) distance in hops, average speed 1m/s **(b)** average migrations delay, 1 and 2 m/s

FIGURE 6 Follow-me efficiency in initial placement and migration events for 240 mobile nodes with individual VMs

During the initial VM placement, that happens when the node first asks for a mobile service (at the beginning of the simulation), all services are optimally placed, making them directly accessible via the AP used by the mobile node, see Fig. 6 a, time = 0.1s. Considering the figure shows the case of 240 services, more than 75% of the available resources in the fog are used. We can then conclude that the initial placement performs extremely well.

After the initial placement of VMs, the mobile node starts its mobility, which is clearly reflected in the simulation logs with the amount of migration events generated in the first 100s of the simulation. The characteristics of the chosen mobility model (Randomway point) are such that, once nodes start to move, a large number of devices will cross the boundaries of the area covered by the community that hosts their VMs. As nodes move in straight lines, they may need to pass through one or more communities on the way towards their randomly chosen destinations, than once they are reached, the mobility process starts all over again.

We have observed from the results provided that in an environment that has a lower number of concurrent migration events, most of VM migrations will successfully follow the node mobility continuously, while providing the lowest possible latency. In the case of an extreme number of migrations, such as the one presented in time = 100s, the successfulness in the migration process is not as high simply because of the fact that it is very difficult to find available resources in each target community in the case when over 75% of the resources are already in use. The results presented with 1, 2 or 3 hops in Fig. 6 a are actually distances in hops between the mobile node and its VM resulting from unsuccessful migration attempts as, according to the migration strategy, only optimal migrations are allowed. Another observation must be made on the influence of the fog layer network topology that also affects the number of hops. Due to the use of a standard fat tree, see Fig. 4 , sometimes there are events when the mobile node moves to a neighbouring community as seen geographically in the physical area, but actually its serving VM needs to be moved to a completely different branch of the fat tree. In case this migration can not be done because there are no available resources in the new destination community, this small movement results in a much higher number of hops between the mobile node and its VM.

Fig. 6 b shows the VM-node delay performance over time for migration events considering 1 m/s and 2 m/s node's average speeds. We have depicted four horizontal lines for the different hop distances, being the bottom one the optimal case when the mobile node application is being served from a server connected directly to the AP that is being used. The upper lines represent the expected delay perceived from the mobile node as the fog VM is placed on a server that is 1, 2 or 3 hops away from the connecting AP, correspondingly. The resulting average delay that is shown in Fig. 6 b is very close to optimal values, with the largest number of fog services being placed the closest to their connecting APs. We can observe a peak at 100s that is related to the high number of migrations performed in the first 100s. The obtained average delay reaches the first higher level of *1 hop* when mobile nodes move at 2 m/s.

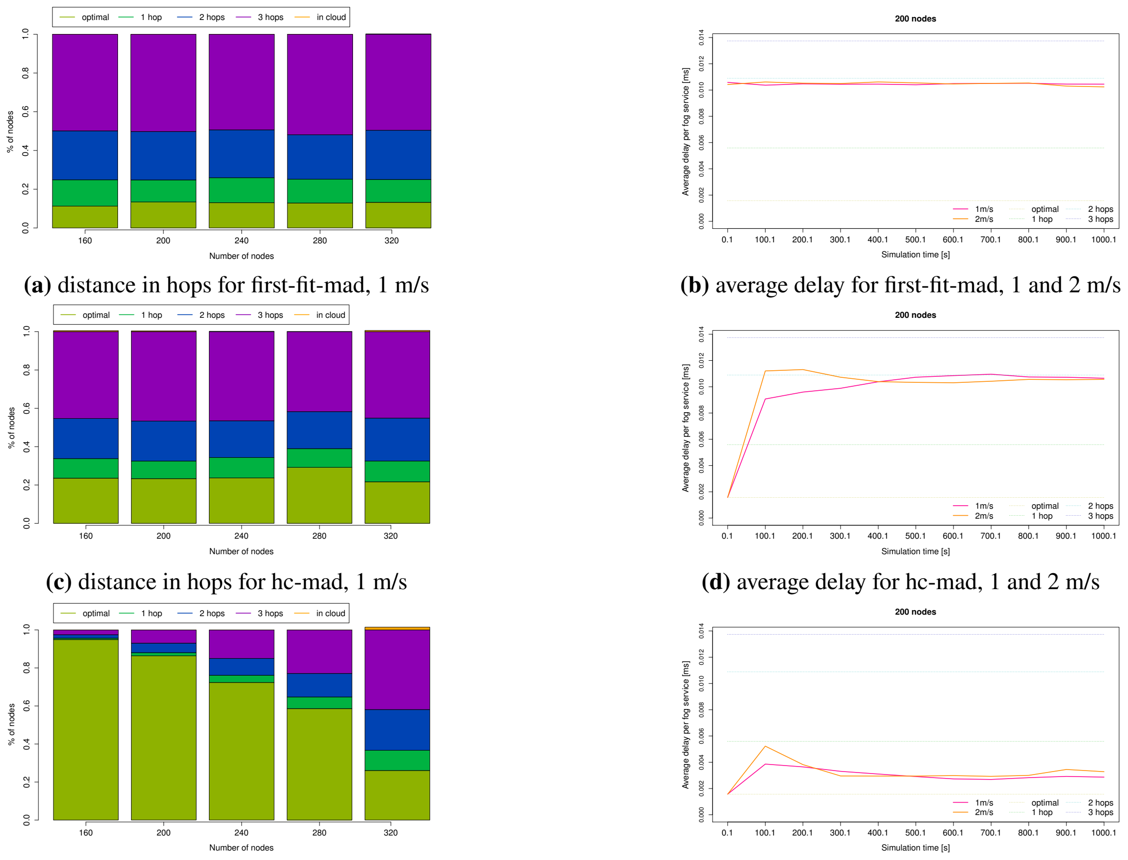
The presented follow-me efficiency scales very well for even larger scenarios. Examples include covering a larger geographical area using additional APs and edge servers while keeping the same radio range, as well as using micro and nano cells for urban 5G networks where the geographical area stays the same but the number of used APs is increased in order to provide full coverage using a shorter radio range. It is worth noting that in the case of short radio range examples, the number of migrations occurring in the system as a whole increases rapidly due to the frequent changes of cells by the mobile users. This causes an additional stress to the backbone network due to a continual large traffic flow incurred by the migration events. In these cases, the SDN based optical backbone must intelligently handle the increased traffic in order to optimally use the available network capacity. However, the introduced hierarchical set of communities helps localise the traffic as much as possible.

4.4 | Efficiency comparison to traditional algorithms

In order to compare and analyse the performance of the proposed fog resource management approach, the identical simulation scenarios have been run using traditional techniques for initial VM placement and migration. First-fit has been used as the most common traditional approach to VM placement where VMs are being placed on edge servers as the requests for instantiation are being processed, using a bin packing technique that is not location aware. For the traditional migration approach comparison we have chosen the *median absolute deviation (mad)*[?] , that aims to provide the optimal consolidation of the edge servers in use which results in maximum power savings. The process is invoked regularly every 100s and it attempts to migrate all VMs from the underutilised servers in order to shut them down. Both, the first-fit and the mad implementations are available as VM placement and VM migration policies in CloudSim. For the cross comparison purposes we analysed the following combinations of policies:

- **first-fit-mad** - initial placement is implemented using *first-fit*, while migration policy is implemented using *mad*;
- **hc-mad** - initial placement is implemented based on our proposal to use *hierarchical communities* mapped to coverage areas, while migration policy is implemented using *mad*; and
- **hc-fm** - the full implementation of the proposed fog VM resource management component: initial placement is implemented using *hierarchical communities*, while migrations are triggered based on the *follow-me* behaviour. The lower layer algorithm employed to choose an edge server from a given community is load balancing (using the server consolidation option has a similar impact on the obtained results).

Fig. 7 presents a graphical cross comparison of the efficacy of the analysed scenarios (first-fit-mad, hc-mad and hc-fm). The total distance in hops and the average delay are presented for different average speed of the mobile nodes. As it is expected, the first-fit-mad combination is the worst performing one with around 50% of the nodes being continuously located furthest away from their corresponding VMs, Fig. 7 a. In this case, only an average of 10% of the nodes are optimally served. The situation is somewhat better for the case of hierarchical communities placement, hc-mad, where the optimally served nodes are doubled to over 20% as an effect of the initial smart placement, see Fig. 7 c. It is worth noting that the performance of the two scenarios does not change when increasing the number of mobile nodes, or the number of fog service, i.e. the load. This is a result of the very rare use of the migrations policy. A very small number of migrations occur throughout both types of simulations because the median deviation policy is rarely triggered since most of the edge servers are already consolidated. Compared to our proposal, hc-fm, the nodes that are optimally served are more than 90% in the case for 160 nodes, going down to 30% when increasing to 320 nodes, see Fig. 7 e. It is clear that the efficacy of the hc-fm approach depends on the current usage of resources in the fog infrastructure. This dependency is directly related to the large number of migrations, which are invoked in order to implement the follow-me behaviour that provides minimum latency between the mobile node and its VM. And with the increased number of



(a) distance in hops for first-fit-mad, 1 m/s

(b) average delay for first-fit-mad, 1 and 2 m/s

(c) distance in hops for hc-mad, 1 m/s

(d) average delay for hc-mad, 1 and 2 m/s

(e) distance in hops for hc-fm, 1 m/s

(f) average delay for hc-fm, 1 and 2 m/s

FIGURE 7 Overall efficiency in providing low latency communication between the mobile node and its serving VM for a varying number of nodes

fog services, the amount of available resources that can be used for migrating VMs is decreasing. However, even under extreme stress with 320 VMs, where a small number of VMs need to be placed in the cloud due to the lack of resources in the fog, the obtained results are still somewhat better compared to the previous two settings (first-fit-mad, hc-mad).

The right hand column of Fig. 7 represents the corresponding overall average delay for each combination of placement-migration techniques for the scenarios when 200 mobile nodes are moving at speeds of 1 and 2 m/s. While the first-fit-mad scenario provides almost constant average delay over time and it is very little dependent on the speed of the mobile nodes since it is completely unaware of the node position, the hc-mad scenario clearly shows the benefits of using a hierarchical community approach during placement, starting from the minimum possible delay and then experiencing worsened performances as simulation time increases because the node mobility causes them to leave the starting physical area mapped to the chosen community where the corresponding VMs reside. For a smaller speed of 1 m/s, it takes about 500 seconds to reach the worst delay that equals to the first-fit-mad case. However, if nodes are moving faster, delay conditions are worsening much faster as well. In comparison, the proposed hc-fm approach enables reaching almost 4 times better results, showing steady performances for the delay after the initial strong mobility that causes a large number of migrations. The performances are a little worse for faster nodes in the beginning only, afterwards the two delay functions for 1 and 2 m/s converge, showing that the proposed VM resource management copes very well with different node speeds being resilient to the faster changing environment.

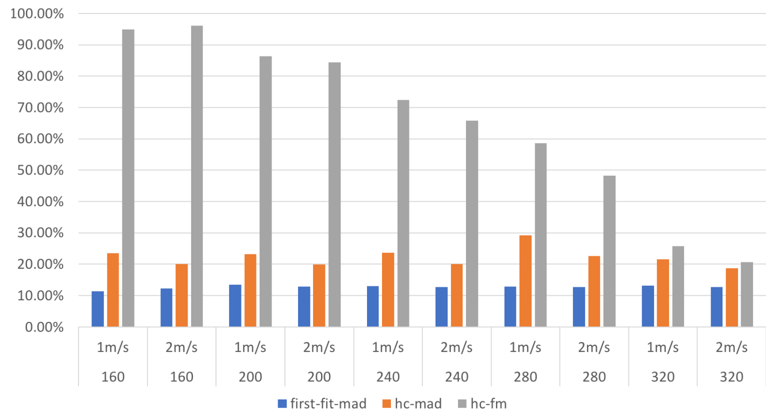
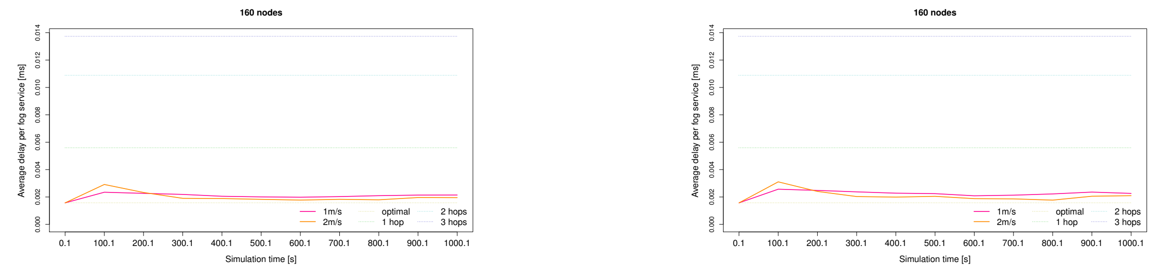


FIGURE 8 Optimality of VM placement comparison depending on number of nodes and their average speed

In Fig. 8 the efficiency of the analysed algorithms is presented using a head-to-head comparison while changing the number of mobile nodes and their average speed. The efficiency is presented as the percentage of node-VM pairs that are provided with the optimum lowest latency possible. The results in the figure are a very clear indicator of the improvement in obtaining high QoS for the end users when using the proposed resource management approach based on hierarchical communities and continuous follow-me based migrations. In addition to the improvement in performance potential, the presented results provide means of gauging how the increase of number of nodes or speed affect the efficiency. It is evident that for very high loads that consume over 75% of the resources available in the fog infrastructure the improvement is minimal.

4.5 | Impact of mapping strategy

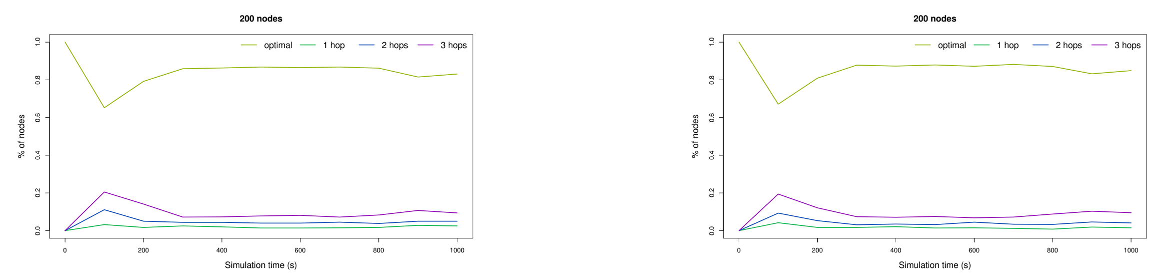
The results presented in the previous subsections are based on scenarios wherein Mapping 1, see Fig. 5 a, is used for creating the relationship between the coverage areas and the logical communities. The consequence of using Mapping 1 can be observed in the cases when a given mobile node changes side of the terrain relative to the vertical center. Based on the mapping, this small change from left to right side (or vice versa) of the terrain results in connecting to a new AP that is furthest away from the currently placed VM, see Fig. 4 , incurring the maximum latency penalty possible in the fog network. If the load in the fog is high, and there are no available resources to perform a successful migration, this handover will result in a significant increase of latency. Based on this reasoning, it is worth investigating whether using different type of mapping can have a different impact on the overall performances of the proposed solution. For these purposes, the same set of simulations where run using a second type of mapping, Mapping 2 as described in see Fig. 5 b.



(a) Mapping 1 **(b)** Mapping 2

FIGURE 9 Average delay in mobile node-to-VM communication for 160 mobile nodes with individual VMs

The overall average delay between the mobile nodes and their corresponding VMs for the two mappings presented in Fig. 9 shows that the choice of a mapping function for physical areas and logical communities of edge servers does not have any significant influence on the fog service performance. To analyse the problem in more detail, Fig. 10 presents the percentage of mobile nodes that are optimally connected to their corresponding fog service, otherwise the distance in nomber of hops is indicated. The example scenario presented is in the case of a higher load with 200 mobile nodes moving with an average speed of 2 m/s. However, even under higher load and speed, the results show that the choice of mapping has no effect on the efficiency of the hc-fm approach. Based on these results, we conclude that the effects of the choice of mapping are not visible on the macro scale and they need to be further investigated by analysing the results on a lower level of granularity, looking into the effects of mobility and migrations inside each independent community.



(a) Mapping 1 **(b)** Mapping 2

FIGURE 10 Distribution of distances between mobile nodes and corresponding VMs

4.6 | Energy efficiency

The last macro analysis is done in order to investigate the power consumption of the proposed solution. The goal is to observe the change in power consumption as the load in the fog increases, the change in power consumption due to different mobile node speed, and how it relates to the traditional techniques that are implemented with the main goal of using a minimum number of servers for hosting VMs in order to improve the energy efficiency. No matter which type of placement or migration is used, the edge servers employ dynamic voltage frequency scaling (DVFS)[?] in order to save energy by lowering the working frequency or parking the cores whenever the full resources of the server are not necessary.

In Fig. 11 and Fig. 12 , the total power consumption of the fog edge servers infrastructure during the simulations is presented for the cases of mobile nodes that are moving with 1 m/s and 2 m/s. Although there are differences in the number of migrations for the cases of faster and slower nodes, while each migration represents additional power consumption because of the additional resources that are being used during the live migration process, the overall power consumption is very slightly affected. As expected, the power consumption increases with the number of nodes since more VMs need to be hosted by edge servers effectively increasing the number of active, powered on, servers.

When comparing to the other placement and migration techniques (first-fit-mad and hc-mad), it is evident that the hc-fm approach incurs extra power consumption. This additional energy that needs to be spent in the fog infrastructure is due to: (1) the higher number of migrations compared to the other techniques, as well as (2) the usage of additional edge servers because in the hc-fm case the balancing of resource consumption that leads to lower power consumption is done only on the local level of one logical community. In essence, this extra spent energy is the price to pay in order to be able to disperse the VMs on a larger number of edge servers so that the mobile node - VM latency is minimised. However, it must be noted that this increase in power consumption is relatively small, perceiving a small increment of 5% of the total power consumed by first-fit-mad

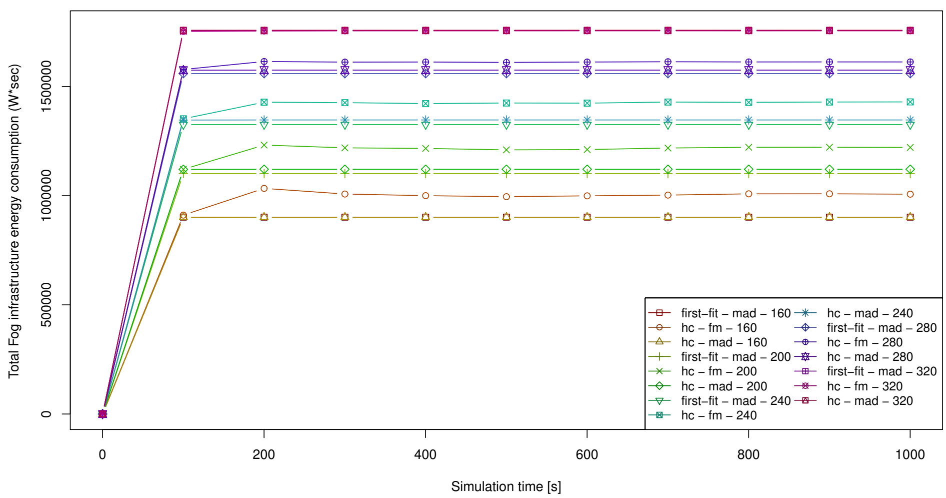


FIGURE 11 Energy consumed by the fog infrastructure during the simulation, speed 1 m/s

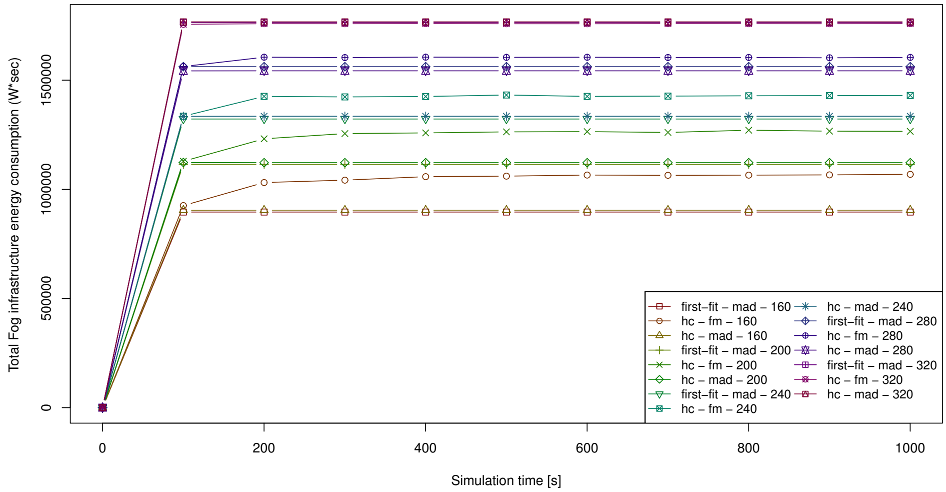


FIGURE 12 Energy consumed by the fog infrastructure during the simulation, speed 2 m/s

5 | CONCLUSION

Focusing on the fog computing paradigm, this paper discusses the problems of efficient resource management in fog layers that will enable a reduction in the expected low delay by implementing location-awareness of the mobile node. For these purposes, the orchestration of the VM resource management module is an essential part of the fog platform when integrating it with the location and mobility tracking module and the QoS monitoring module.

Aiming to detail the operation of the resource management module, a hierarchical community-based approach has been proposed for solving the problems of initial placement of the individual VMs of mobile nodes so that they are located in the logical communities of edge servers that correspond to the physical location of the node. For the purposes of continuously providing the lowest latency between the node and its VM in a mobile environment, the resource management module is activating a migration algorithm that effectively implements a follow-me behaviour for the end user. The live migration is triggered whenever a mobile node starts a handoff process by moving from one geographical service area to another, in which case the corresponding VM needs to be migrated to a different logical edge servers community.

The presented results confirm that very high accuracy in the follow me efforts for the fog services can be achieved. Using the proposed hierarchical community technique, the initial allocation of fog VMs is almost optimally performed when considering a low, medium and even heavy loaded fog infrastructure. In the results section we analyse the migration events and migration attempts occurred during the simulation and we observe how the mobility-aware fog migration implementation manages to keep the average service delay very close to the ideal optimal value, and while exhibiting dependency on the system load and mobility speed, it still manages to perform convincingly in stressed environments with continuously fast moving walking users. The comparison analysis shows that the proposed approaches for initial placement and migration are providing significant improvements in the achieved average delay compared to traditional algorithms, with a slight increase in the energy consumption due to a higher number of active edge servers. While these initial results show that the node speed influence can be observed only for very high loads, and the mapping of physical areas to logical communities has almost no effect on the observed macro performance, a more in-depth analysis is needed in order to fully investigate the performance dependencies on these parameters.

Our future work will focus on analysing the changes that occur during simulations on the community and single node levels, as well as the amount of network traffic generated because of the activated migrations. In addition, we are interested in research a broader range of scenarios involving more complex fog services with multiple VMs while combining the network configuration with SDN in order to implement the proposed solution in a test environment.

How to cite this article: Filiposka S., Mishev, A., and Gilly, K. (2018), Mobile-aware Dynamic Resource Management for Edge Computing, *Trans Emerging Tel Tech.*, *2018;00:1–6*.