Supporting Location Transparent Services in a Mobile Edge Computing Environment

Katja GILLY[1], Sonja FILIPOSKA[2], Anastas MISHEV[2]
[1]*Department of Physics and Computer Architectures, Miguel Hernandez University, Elche, Spain*
[2]*Faculty of Computer Science and Engineering, Ss. Cyril and Methodius University, Skopje, Macedonia*
katya@umh.es

***Abstract*—Emerging models such as mobile edge computing provide the necessary characteristics for the deployment of Internet of Things applications by supplying the connected devices with local computing facilities essential for latency sensitive applications. One of the major issues of the underlying edge computing architecture is to cope with the device mobility that imposes dynamically changing network requirements. In this paper, we propose a resource management approach that aims to improve the location transparency and provide high quality of experience for end users by optimising latencies perceived when nodes are accessing services hosted on the edge of the network. By managing the virtualised computing resources based on the node location area information, the main objective of the approach is to minimise the network latency perceived by mobile nodes for both the initial allocation and the dynamic resource migration during the service lifetime while the requester node is changing location areas. The system is trying to achieve the most accurate 'follow me' service where the assigned resources closely follow the current mobile node location. The presented results show the effectiveness of the proposed solution in comparison to traditional resource management techniques on the macro and micro scale.**

***Index Terms*—context-aware services, handover, mobile nodes, resource management, wireless networks.**

I. INTRODUCTION

The imminent deployment of 5G that will enable the implementation of a network architecture supporting the convergence of various network systems under one umbrella is a step forward to the development of future network services with a wide range of requirements. Using the paradigms of Software Defined Networks (SDN) and Network Function Virtualisation (NFV) as enablers for a flexible, programmable network design that will provide end-to-end slices with different Quality of Service (QoS) on demand, the 5G architecture is an environment that supports high device density and mobility, high capacity and low latency [1].

This makes it a viable solution for an overarching implementation of new services that have strict demands that go beyond the network QoS and include substantial processing power for providing real-time feedback to users such as on-the-fly high quality 3D rendering for supporting augmented reality applications. For these purposes, the network needs to be expanded with mobile edge virtualised computing facilities that are positioned at the network edge, co-located with the mobile base stations and wireless access points. In the cases of context-aware services and real-time applications that depend on the user location, it would be most efficient to provide the computing capabilities on the network edge where the user can access them via only one hop provided by a wireless access point (AP) [2].

Following this approach, the overall system will provide end-users with local computing power that can be supplied on demand together with the request for specific network QoS, thus supporting new innovative service that demand both network and computing power and achieving high quality of experience (QoE) for the end users. Apart from an improved user experience, the additional computing power can also be used for adding enhanced features to the network capabilities. Based on NFV and automation it can provide adaptive network behaviour which supports agile services and scalable solutions [3].

Several active attempts have arisen recently aiming to model the architecture of a general multi-access edge computing (formerly known as mobile edge computing – MEC), standards and deployment platforms, such as MEC as defined by ETSI and the OpenFog initiative [4]. They support the development of the 5G ecosystem extension with edge computing by setting the requirements, underlining the main use cases, and defining the necessary modules. The general MEC architecture includes the mobile edge host that hosts the edge computing facilities and the mobile edge platform responsible for hosting the mobile edge services. The mobile edge applications run as virtual machines (VMs) on top of the virtualisation infrastructure provided by the mobile edge host and interact with the platform to provide the mobile edge services.

The mobile edge orchestrator is the core functionality for managing the mobile edge system. This entity is responsible for maintaining an overall view of the system based on the deployed mobile edge hosts, used and available resources, topology, etc. This entity is also responsible for selecting the appropriate mobile edge host for application instantiation based on the provided requirements such as computing power and latency, but also for triggering relocation of the application when necessary to support the requirements set.

In this paper, we focus on the set of functionalities provided by this orchestrator that deals with dynamic resource management of the mobile edge hosts that are geographically distributed throughout the provider network. The main goal is to optimise the available virtual resources based on user demands. In contrast to cloud computing, the mobile edge specific resource management problems are under the influence of the dynamically changing position of the mobile users. In order to ensure the minimum latency, the placement of the virtualised resources relative to the

Digital Object Identifier 10.4316/AECE.2018.04002

available physical edge hosts has to be relative to the AP the user is connected to. This makes the resource management policy for edge computing particularly sensitive on the geographical location of consuming nodes. In order words, the main goal is to implement a dynamic behaviour that will decide when to trigger live VM migration in order to ensure that the location of the mobile edge application will be as close as possible to the service consumer.

The resource management problem for the edge computing is becoming the focal point of recent research [5]. However, while the problem is fully recognised, and the performance of resource allocation techniques have been analysed, the efficient continuous monitoring and migration triggering techniques are yet to be studied in detail. In order to contribute to this field of research, this paper focuses on the problem of mobile edge hosts dynamic resource management. Both placement and migration management are based on the properties of the 5G hierarchical network architecture creating a dendrogram of the wireless service (geographical) areas as they are perceived by the mobile edge deployment. The presented results shed light on the performance of the location area handovers for the mobile edge applications triggered so that the mobility pattern of the wireless device is closely followed.

The remainder of this paper is structured as follows: Section II introduces the characteristics of the mobile edge resource orchestrator and its components. Section III describes our proposal for location-aware mobile edge orchestration detailing the initial resource allocation method and the migration procedure when handoff occurs. Section IV provides the implementation details, to subsequently focus on the simulation results presenting the efficiency of the approach compared to traditional techniques and detailing the influence of different parameters such as the number of mobile nodes and the speed of the mobile nodes. Finally, Section V concludes the paper.

II. Orchestrating Virtualised Resource Management on the Edge

By adding computing resources to the new generation radio access network site, the capabilities are extended with the possibilities to provide virtualised computational capabilities to the mobile nodes. This concept has been initially presented in [6] by Bahl, where the “micro datacentres” design is based on wireless cells that are augmented with local computing power provided by a small number of co-located computing resource nodes, so called edge servers (see Fig. 1). In order to be able to orchestrate the resource management of these edge hosts centrally via the mobile edge orchestrator, the distributed mobile edge hosts need to be connected using a virtual overlay network that can be defined and maintained using NFV and SDN. The work presented in [7] and [8] discuss how these technologies can be implemented and their capabilities for supporting mobile edge computing. In essence, NFV and SDN can be used to address the effective operation of the fundamental technical requirements in the MEC architecture including dynamic orchestration of the processes for allocation and migration of virtualised resources in the network provider infrastructure that encompasses the core network and the edge mobile broadband networks.

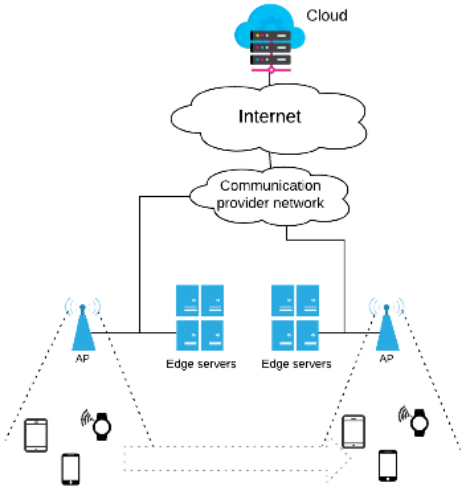


Figure 1. Overview of the MEC architecture

The implementation of the MEC layer as an extension of the 5G architecture requires the deployment of management on multiple layers: system level management via the mobile edge orchestrator, cell level management via the mobile edge platform manager, and host level management of each individual server.

Mobile nodes may request to use MEC provided services from the mobile edge orchestrator which is going to decide to which platform manager to forward the request, that is then going to instantiate a VM to host the requested mobile edge application on a host with enough free resources. All resources available on the mobile edge hosts are virtualised and centrally managed by the mobile edge orchestrator, while the platform managers are ensuring local optimisation and load balancing. For the orchestrator to be able to decide on the VM placement, it must be implemented as a combination of several interacting components that are part of the system level management and monitoring, as it is shown in Fig. 2.

The *location tracking module* provides the current location of each mobile node, assisting the decision for seamless handoff from one wireless cell to another one. This information is also to be used as input for the *quality of service (QoS) monitoring module*. This module is responsible for constant monitoring of both the quality of service in terms of connectivity and the resource usage of each service and mobile node. Depending on the mobility events that are received from the location tracking module, the QoS monitoring decides whether the VM that is providing an edge service needs to be migrated to the edge servers hosted in the new cell in order to ensure continuous low latency. The *VM management module* is responsible for the management of VM virtualised resources. It implements the placement and migration techniques that are used to find the optimal edge host where the VM can be hosted, constrained by the resources usage current status and requirements obtained from the QoS module. The *cost analysis module* tracks the usage of mobile nodes' assigned resources. Another function of the cost analysis module is the selection of the best cloud service provider when additional computing resources are necessary in the cloud. The *security and identity module* is used to provide security features by implementing identification processes based on

credentials, enforcing policies, and activating accounting for cost analysis. The *communication service* is the *module* that communicates with the mobile edge platforms receiving resource demands and any other MEC related events.

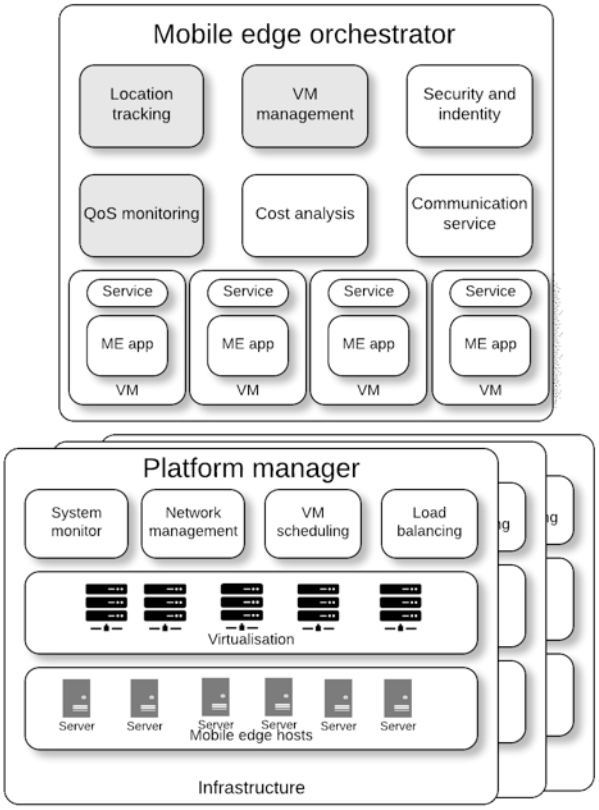


Figure 2. MEC architecture components

As discussed, (see Fig. 2), the proposed dynamic resource provisioning in mobile edge hosts is defined using components within the high-level mobile edge orchestrator. This approach enables network wide efficiency when provisioning services, where the monitoring system in charge of collecting and reporting on the behaviour of resources is used as a control feedback mechanism. The main functionalities that need to be implemented by the orchestrator is initial resource allocation for newly requested edge services, and migration of existing services that ensure maximum QoS throughout the service lifetime.

Upon analysing the system architecture, it can be concluded that the problem of placing a VM in the edge infrastructure can be solved hierarchically: The first level optimisation in the decision-making process is focused on achieving as minimal latency as possible, i.e. to find the required resources at the closest cell to the requesting node. The best-case scenario is to choose the mobile edge platform that governs the mobile edge hosts co-located with the current cell that the mobile node is connected to. The second level of optimisation is then used by the selected mobile edge platform in order to select the most suitable mobile edge host that is available on the platform. This choice can be made based on different goals such as resource load balancing or energy-efficient consolidation.

Efficient migration implementation is of utter importance due to node's mobility. Without implementing migration of the hosted services, the effect of optimal initial resource allocation will be nullified as soon as the node starts to move across cells. Thus, the migration of virtualised resources needs to be triggered as a response to the handoff events occurring in the network. While the initial resource allocation problem has some similarities to the resource allocation problem in cloud computing, the resource migration problem is very different because, in this case, the main goal has nothing to do with resource consolidation, but to the contrary, it is only concerned with maximum QoS. This requires that each time the mobile node is handed off to a new cell, a decision must be made about the migration of the corresponding mobile edge application used. Obviously, the aim should be to migrate the hosting VM to the mobile edge platform providing minimum distance in network hops to the new mobile node location given the current availability of resources.

Thus, the decision process is based on the information about node location provided by the *location and mobility module* combined with the corresponding information from the *QoS monitoring module* that is fed into the *VM management module*. These migration events requests, also known as VM handoffs, have also been studied in a similar fashion by [9], where the functionalities of the necessary components are outlined.

A literature review shows that a number of research papers are focusing on optimising different aspects of mobile edge dynamic resource management, potential algorithms and their efficiency. However, there is still a necessity for a holistic framework approach that will encompass all aspects of mobile edge resource management in a scalable and efficient manner under the assumptions of high mobility and large geographical distribution. One example of the existing proposals can be found in [10] which is based on the concept of cloud atomisation that describes the available virtual resources using a high level of granularity. In this proposal, the decision on placement is strictly based on load balancing the available resources, which is achieved using graph repartitioning techniques. The approach presented in [11] has some similarities to the proposal given in this paper, both aiming to minimise the delay when the user accesses a hosted mobile edge service. The main setback in [11] is that the authors only consider the optimality of the initial allocation of the resources, without employing migrations to maintain this optimality for mobile users, which makes the initially optimal resource usage far from optimum over time. Live service migration in the mobile edge cloud is discussed in details in [12], where a layered framework is presented focusing on the execution of the service migration, a process that is independent of the algorithms used for optimisation of the migration decisions. The migrations methods proposed are an enhanced version compared to the traditional migration techniques used in the cloud, and work for both virtual machines and containers.

Authors in [13] identify the resource management component as a key component of the architecture to minimise the latency and maximise the throughput as one of the most critical problems, especially for health monitoring and emergency response IoT connectivity. Their proposed architecture consists of a resource management component defined with a provisioning and scheduling module, that coherently manages resources in such a way that application level QoS-constraints are met, and resource wastage is

minimised during placement. Cloud-only and edge-ward placement approaches are compared. In [14], using the CloudSim simulator, authors propose a model for resource usage estimation and reservation, together with pricing for IoT users, based on the probability of the fluctuating relinquish probability of the customer, service type and price. A joint resource allocation and carbon footprint minimisation approach is discussed in [15] with a case study of video content distribution and streaming. The authors propose a new distributed optimisation algorithm that maximises bandwidth for all demanded video traffic.

A survey of the concept of service migration in MEC is presented in [16], where a review is given on the live migration techniques used in datacentres and the handover process in cellular networks. The authors discuss the technicalities of migrating a running service based on the amount of data transfer and processing required. They also introduce the idea of the follow-me cloud and propose two heuristics for the migration decision process based on one and two-dimensional Markov decision processes that deal with scenarios where users are moving along a straight line or within a square area. The survey also discusses the possibility of using artificial intelligence for migration events prediction and blockchain for solving the trust issues.

Different Fog service orchestration architectures for the Internet of Things are analysed in [17] for various application scenarios. The ETSI MEC architecture is compared to three other proposed architectures. The authors identify resource management as one of the major research challenges in edge orchestration and discuss the features of the modules that need to deal with scheduling, path computation, allocation and optimisation.

The focus of this paper is the implementation of the QoS monitoring and VM management modules that respond to the changes in location as reported by the location tracking module. Together these three components enable the implementation of a dynamic resource management of the MEC infrastructure that would effectively ensure that the used edge resources closely follow the mobile node across the edge network thus ensuring minimum latency. This follow-me principle is implemented via specialised VM placement and migration algorithms based on the mapping of the cells' location area to the logical groups of edge servers of the corresponding mobile edge platform.

III. Proposed Location Based Strategies for Mobile Edge Orchestration

In this section we describe the orchestration solution we propose to obtain optimal latency times of edge services by defining a location-aware dynamic mobile edge resource management proposal that, in addition to the optimal initial placement of VMs, it continuous to maintain optimality by triggering VM migration whenever lower QoS is detected. With low latency as the main objective, optimal VM placement is considered to be the placement that will provide the minimum latency between the allocated mobile edge application and the requesting mobile node. This proposed solution follows the MEC hierarchical architecture and decouples the proposed algorithms used: one used by the central orchestrator to choose the mobile edge platform, and another used by the mobile edge platform to select a mobile edge host. In this way, the objectives encoded in the mobile edge platform can be based on any goal that is considered important by the provider (energy efficiency, load balancing, etc.) not affecting the overall low latency objective of the central orchestrator. It must be noted that the general proposal outlined in the paper is not dependent on virtualisation type in terms of virtual machines or virtual containers and works equally for both. The remainder of the paper discusses the implementation and results using the VMs virtualisation method since the container migration has not reached the maturity level of VM live migration [17].

The network infrastructure can be represented using a hierarchical tree, where each leaf node represents one cell in the mobile wireless network. The granularity of the representation of the cell depends on the use case scenario and is defined as the set of all radio elements that are governed by the Radio Access Network (RAN) equipment that has its own mobile edge platform, i.e. the set of antennas and radio controllers that are part of the mobile edge network governed by one mobile edge platform. In other words, there is only one set of mobile edge hosts that correspond to the leaf node cell, and these are managed by one mobile edge platform. Each cell has its own location service area, that is the geographical area that is served by the cell, where all mobile nodes that are located in this location service area are connected to this cell. When a mobile node leaves one location service area and moves to a different one, a handover event occurs, that might also trigger a corresponding mobile edge migration event. This mapping of cells to their location service area is the connection between the mappings of the provider network and the physical geographical information, see Fig. 3. In the provider network, all cells are connected in a hierarchical fashion using distribution and core network layers. On the cell level, each mobile edge platform has a well-defined pool of available and used virtualised resources that reside on a number of mobile edge hosts. These resources can be described using different sets of features such as the available memory (RAM), number of computing cores available, network capacity, etc.

As already mentioned, based on the location of each cell, the mobile edge platforms and hosts are seen as servicing a specific service area defined with the wireless coverage of the cell. This service area can be visualised as a projection over the geographical area (terrain) served by the network provider. For an example see Fig. 3, wherein the Voronoi cells [18] represent the service area of each AP and are obtained by assigning each coordinate of the terrain to the closest cell based on the signal strength. If one mobile edge platform is combined by multiple cells, then the corresponding location service area of the platform is the complete area covered by all participating cells, while the mobile edge hosts that are managed by this platform are all mobile edge hosts located within this location service area. In this way, the complete service area of the provider is divided in a set of non-overlapping geographical areas that are mapped using a 1:1 relationship to the mobile edge platforms that govern the corresponding mobile edge hosts.

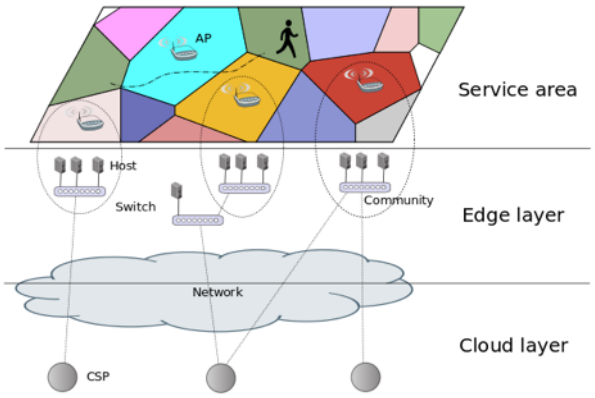


Figure 3. Projection of the mobile edge hosts onto the wireless service area for an example terrain

The system level mobile edge orchestrator implements the high-level minimum latency implementation by taking advantage of this one to one location service area to mobile edge platform mapping. As the mobile node moves from one location service area to another, the VM management module aims to ensure that the mapping stays as optimal as possible. It is important to note that two neighbouring location areas may be served by mobile edge platforms that are multiple hops apart in the logical provider network. This is an effect of the network topology that interconnects the edge nodes of the provider. Typical interconnections of the edge resources follow a tree pattern with the centralised root layer on top that then, divides into distribution areas and finally, spreads to cover distinct location service areas on the network edge level. This means that, as a mobile node moves from one location service area to another, a simple handoff to a neighbouring cell will trigger a request for migration of the corresponding node VM from one platform to another that may be *x* logical hops away.

A. Mobile edge orchestration for initial resource allocation

The problem of VM placement represents a subset of the problem of optimal resource usage [19]. Despite the very high computational complexity, many heuristics have been developed focusing mostly on trying to optimise a given cost function. This function represents the optimisation goal of the heuristics under a set of constraints that define the compatibility of the requested resources for the VM and the available resources in the pool of resources. In the past, these heuristics were confined to describing the resources using one parameter only, exploring the space of possible solutions with one optimisation goal. Recent attempts, however, deal with complex scenarios that use multiple dimensions to describe the set of different resource characteristics (CPU, RAM, OS, storage, etc.) and can focus on achieving optimisations across a set of objectives.

As discussed previously, the optimisation method we define is based on the hierarchical representation of the collection of MEC platforms (and associated APs with their service areas) in the form of a dendrogram which can be achieved with any method for hierarchical clustering. The dendrogram can be described as a symmetric matrix *E*, in which e_{rq} is the lowest hierarchical level at which objects *r* and *q* belong to the same cluster. In the case of optimising MEC service placement, the objects of interest *r* on the lowest level is the access point the requesting user is connected to, while *q* is the edge server host that needs to be selected in order to allocate the necessary resources for the service and obtain minimum latency. On the lowest level the smallest clusters consist of all *q* edge servers belonging to one MEC platform and all *r* APs that are directly connected (on the same switch) to that MEC platform. Going up the dendrogram these objects are grouped into larger clusters.

Let *S* denote the set of services and *N* the set of edge server nodes in the network. Based on *E*, we define the distance matrix $L=\{l_{ij}\}$, where $l_{ij} \in \{0,1,...,H\}$ is the distance between objects *i* and *j* and *H* is the highest point on the dendrogram that represents a cluster encompassing all objects. For the given matrix *L*, find the minimum distance *l*, $l \in L$, if the user is connected to an AP *a* and the requested MEC service is specified with its resource type requirements *RSk* for $k \in \{0,1,...,K\}$. Combined with the second level decision of load balancing, the problem of choosing an edge server can be expressed as the following:

$$\min_{m} \min_{l_{a,n}} \max_{k,n} p_{n,k}(M)$$

where *m* is the mapping function for placement of service *s* to edge host *n*, $m = S \rightarrow N$ and *M* is the total number of services in the moment when the request arrived.

$p_{n,k}$ denotes the aggregated weighted cost for the edge server $n \in N$ for resource type *k* for all successively placed services up to *i*:

$$p_{n,k}(i) = \sum_{j=1}^{i} \sum_{v:m(S)=n} d_{v \rightarrow n,k}(j)$$

and *d* is the weighted cost of assigning service *v* to edge server *n* serving as normalisation to the total resource capacity.

Heuristically described, using the information from the provider network structure and the mapping to the location service areas, we propose that the decision for the initial VM placement based on the current global status of the resources available at the mobile edge servers must be focused on the optimisation (minimisation) of the *mobile node - mobile edge application (VM) latency* which is decomposed into a two-level approach:

1. $\min_{l_{a,n}}(\ )$ - identify the mobile edge platform that is (a) logically closest to the currently used wireless cell and (b) has enough resources available and thus can be used for placement of the user requested service,
2. $\min_{m}(\ )$ - based on the available resources among the mobile edge hosts managed by the chosen platform from level 1: employ a secondary – lower-level allocation algorithm such as server consolidation, or load balancing, or another traditional approach to identify the exact host(s) where the requested service will be placed.

In other words, the initial placement is achieved using a multi-objective function, wherein the higher-level objective is minimising latency and the lower-level objective is load balancing or resource consolidation for energy efficiency (depending on the chosen algorithm). These objectives must be met while working with the restrictions imposed by the current status of the virtual resources usage across the mobile edge platforms, the user location and active service

area, and the interconnecting provider core network. The problem can be translated to a bin-packing problem which can be solved using an optimisation technique based on multi-dimensional vector algebra, where each dimension describes a single characteristic of physical mobile edge hosts and virtual resources (ex. RAM, cores, bandwidth). Moreover, the algorithm for mobile edge platform level resource optimisation is triggered in regular intervals in order to ensure optimal resource usage. In this way, it can transparently rearrange VMs in one group of mobile edge hosts so that higher optimisation is achieved (lower power consumption, or better load balancing) [20].

The most optimal solution implies that the chosen mobile edge platform is going be co-located with the cell where the mobile node is connected. In the worst-case scenario when there are available resources in the edge, the placement algorithm will choose a logically distant mobile edge platform, since the closer ones do not have the available resources. In the case when there are not enough available resources anywhere in the edge network, the service request is dispatched to the upper layer cloud provider.

B. Mobile edge orchestration for resource migration

For mobile nodes that move from one to another location service area, a live VM migration needs to be triggered for each handoff event where the new wireless cell is served by another mobile edge platform.

In other words, we consider that when the mobile node travels the boundaries of service areas and makes a handoff of the wireless connection to a new cell operating in conjunction with a different set of mobile edge, a resource migration event must be triggered in order to preserve optimality in terms of minimum latency. This event is recognised with the cooperation of the location tracking module and the QoS monitoring module that are triggered during a handoff. If the change of cell translates to change of the currently used mapping of location areas to mobile edge platforms, then the VM management module is informed and an attempt for migration is triggered.

The core purpose of our migration process is to maintain the highest QoS in terms of latency as possible under the current conditions in the network. Thus, whenever a node moves into a new service area, the QoS module is triggered with the higher latency occurring between the node and the corresponding mobile edge application. In order to maintain the low latency conditions, the migration algorithm attempts to locate the mobile edge platform that is related to the new cell and check if there are resources available to support the migration process. If the mobile edge platform serving the new cell has enough resources, the live VM migration process is triggered and the destination mobile edge platform employs the second level optimisation algorithm to choose where exactly to place the migrating VM.

In the case when the second level optimisation algorithm reports that the VM cannot be migrated because of a lack of resources at the destination platform, migration is cancelled, and the mobile edge application stays where it was originally hosted. By implementing this behaviour, the unnecessary use of additional resources for migration is avoided since any other migration would effectively entail the same or longer latency when compared to the decision not to migrate the VM. This reasoning if further analysed in the subsequently presented results. However, if migration is triggered and the mobile edge application is currently hosted in the cloud (no available resources were found in the edge network during initial placement), the migration process will behave exactly as the initial placement algorithm, continuing to search for a suitable mobile edge platform aiming to migrate the mobile edge application bringing it closer to the end user. Migrating the VM anywhere in the lower edge layer will significantly increase the quality of experience for the user.

The process of migration must be strictly implemented using live migration ensuring that the mobile node does not lose any connectivity to the mobile edge application during the migration process. While this ensures high QoE from the mobile node perspective, it requires additional use of the mobile edge hosts resources since during the migration the required pool of virtual resources for the migrating VM is consumed on both the source and destination mobile edge host. After the full migration process is completed, the source mobile edge host will release the resources that belonged to the migrating VM. This must be considered when dimensioning the resources in the MEC infrastructure.

IV. Implementation and Simulation Results

We have implemented several components of the MEC platform reference model previously discussed in Section 2 by extending the popular CloudSim simulator [21].

The complete edge infrastructure is implemented by adding a MEC layer consisting of mobile edge hosts and functionality components that implement the logic of the mobile edge orchestration and mobile edge platform managers for each set of mobile edge hosts. This MEC layer framework is then connected to a traditional cloud datacentre to represent the higher cloud layer.

All computing nodes and mobile edge hosts in the cloud and edge layer, respectively, are characterised using three dynamic magnitudes: number of cores, size of RAM in GB and available network bandwidth in Gb. The network switches and routers are used to represent the hierarchical interconnectivity network that connects the mobile edge platform infrastructures. The typical edge, distribution, core layer network architecture is used in a fat-tree like manner with a 1 Gb network bandwidth for the edge links, and 10 Gb bandwidth for the upper distribution and core layers. This scenario can be effectively configured in the form of an overlay virtual network that can run on top of the communication provider physical network infrastructure. Each edge network node connects a set of mobile edge hosts that are governed by one mobile edge platform manager and are co-located with at least one wireless AP. The wireless APs that are part of one mobile edge platform manager define the location service area, i.e the cell serviced by the manager. The connection to the cloud is done via a wide area network link from the core layer generating the maximum distance from wireless mobile nodes. The resource management framework implementation is completely independent and can be used with other network provider topologies enabling enhanced flexibility in the creation of simulation scenarios cross comparison studies.

CloudSim offers a small set of VM placement and

migration policies that expanded with an additional set of MEC VM placement and MEC VM migration implementations. These additions are divided between the mobile edge orchestrator, implemented in the VM management module, and the mobile edge platform manager, implemented in the VM scheduling module. The higher-level set of policies decides on the mobile edge platform manager that is going to be invoked, while the lower level set of policies at the platform manager decides which mobile edge hosts are going to host the VMs.

At the mobile edge orchestrator, the MEC VM placement is called at the first request for a VM by each mobile node in the simulation, and the VM migration process is triggered as a reaction to received external migration request events created by the location tracking and QoS monitoring modules. At the mobile edge platform manager, the MEC VM placement is invoked by a request from the mobile edge orchestrator, and the VM migration process is regularly checked in order to provide constant resource optimisation.

The location tracking module is controlling the complete provider-wide service coverage area, the current location of all mobile nodes and the sites of the wireless APs. A node mobility model or a real-life mobility trace file describing the pattern of movement of each node can be used as an input for the mobility events to the module. The module creates a handoff event every time a mobile node changes its wireless connection to a new AP.

The QoS monitoring module receives the handoff event information and analyses whether migration is needed by determining if the change of AP has resulted into a change of the location service area that is mapped to a different cell, and hence, a different mobile edge platform manager, compared to where the mobile node serving VM resides. In this case, it sends a migration event to the VM management module activating the migration policy.

A. Scenario description

Having in mind to thoroughly analyse the performance of the proposed initial placement and live migration strategies, we have implemented and tested a set of scenarios using the described CloudSim extension for edge computing.

Each MEC service is offered by an individual VM for each mobile node. The requested service / VM size is randomly chosen from the following possibilities: mini (1 core, 1 GB RAM, 100 Mb), medium (1 core, 2 GB RAM, 100 Mb), large (2 cores, 2 GB RAM, 100 Mb). The MEC layer is interconnected using the described layered hierarchy, where at the edge layer there are 10 mobile edge hosts per mobile edge platform manager co-located with 2 wireless APs that define the service area for the mobile edge platform manager. The total coverage area is divided into 8 service areas governed by 8 mobile edge platform managers with 80 hosts in total. Higher-level interconnection between mobile edge platform managers is achieved by employing additional distribution and core layer routers in a symmetrical fashion for redundancy.

In essence, the provider's virtual overlay network is designed as a symmetrical fat-tree encompassing the complete MEC layer where the potential distances between a mobile node and the corresponding VM(s) can be described as follows:

- *optimal*: the mobile node is communicating via the wireless AP that belongs to the mobile edge platform manager where the serving VM is running.
- *1st level hierarchy*: the mobile node communicates with the service VM(s) via additional network nodes from the distribution layer, because the VM is located on a server that is part of a neighbouring mobile edge platform manager.
- *2nd level hierarchy*: the communication is achieved via one core layer device, because the service VM is hosted by an edge server belonging to a mobile edge platform located in a different network branch.
- *3rd level hierarchy*: the communication can be achieved only via multiple core layer devices, because in this case the service VM is hosted in the furthest part of the network tree.

The latency given in the results subsection is calculated based on the available processing delays defined in CloudSim [21] for edge, distribution and core routers respectively, see Table I.

TABLE I. PROCESSING DELAY IN NETWORK ELEMENTS

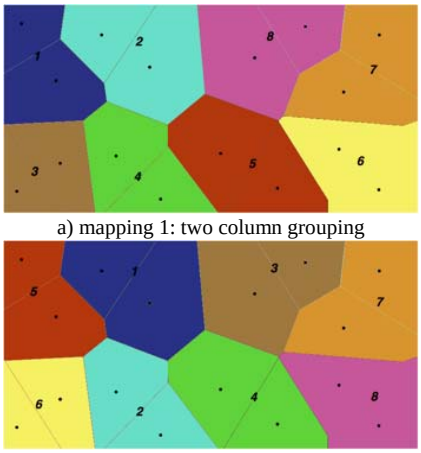
Level	Processing delay
Edge	0.00157 s
Aggregation	0.00245 s
Core	0.00285 s

All mobile edge hosts are specified with 12 GB RAM, 6 computing cores and 1 Gb network bandwidth. The implementation of the VM placement and the VM migration policies for the mobile edge platform managers includes two choices for the lower-level optimisation: vector-based load balancing or vector-based server consolidation [22]. The results provided are obtained using load balancing only.

The terrain that represents the network provider complete coverage area is defined as a 400 m x 800 m rectangle that is divided into 8 roughly equally sized cells, positioned in 4 columns by 2 rows. For each cell there are 2 wireless APs with randomly chosen locations. Based on the random locations, the service area for each AP is obtained as a Voronoi cell (see Fig. 3) where the maximum range of each AP is set to 200 m. The 8 platform manager cells are then mapped to each coverage area. Two possible mappings have been considered so that the effect of different mappings can be analysed. Fig. 4 represents the two types of mapping used, where the dots represent the coordinates of the APs, and the colors and numbers correspond to the mobile edge platform manager mapped to the given physical area. The mappings have been chosen so that, in the first case, the first 4 and second 4 mobile edge platforms belong to different parts of the interconnecting network, while in the second mapping the central 4 are interconnected via one core layer network node, and the first and last column platforms belong to the second part of the network that shares a different core layer network node. For the results that follow, the scenarios that use Mapping 1 for describing the relationship between the physical areas and the mobile edge platform are presented unless stated otherwise.

The location tracking component is in charge of generating the events for the initial allocation and the succeeding migration events in accordance to the information provided by a mobility trace generator. The

mobility trace is generated using a Random Waypoint mobility model where a varied number of mobile nodes (160, 200, 240, 280 and 320) move with an average velocity of 1 or 2 m/s corresponding to a leisure and faster walk.



a) mapping 1: two column grouping

b) mapping 2: center and edges grouping

Figure 4. Mapping physical coverage area served by two co-located APs to mobile edge platform managers

With the aim to investigate the efficiency of the proposed policies for edge services placement and VM migration, five different sets of simulation traces are analysed grouped according to the number of mobile nodes. Each node is assumed to request one individual VM in the edge layer. Based on the capacity of the defined mobile edge hosts, once all nodes request a VM, the total resource capacity of the edge layer is used up to around 50% for 160 nodes, 75% for 240, and 100% (with less than 1% VM hosted in the cloud) for 320 mobile nodes. The scenarios have been chosen intentionally so as to analyse the behaviour of the system when under strain (over 50% usage) with and without enough available resources to manage successful migration events needed for effective follow-me behaviour.

B. Cross comparison with traditional approaches

To gauge the effectiveness of our proposed resource management algorithms, a cross comparison study has been made by running the same simulation scenarios wherein a different technique is used for the initial placement and/or migration. The traditional initial placement technique used for comparison is first-fit, where VMs are sent to the mobile edge platform managers and then, packed on mobile edge hosts as they arrive without any awareness for the location of the mobile node that requests the VM. The traditional migration technique used for comparison is the median absolute deviation (mad) based migration [23], where migration events are triggered in order to achieve the best consolidation of used edge hosts, aiming to achieve maximum power savings. Both techniques are readily available in CloudSim.

In effect, three different combinations were analysed:

- **first-fit-mad** where initial placement is done using *first-fit* and migrations are made according to the *median absolute deviation (mad)* based migration;
- **hc-mad** where initial placement is done using our proposed technique based on *hierarchical coverage areas* and migrations are made according to the *median absolute deviation (mad)* based migration; and
- **hc-fm** where our fully proposed VM resource management for MEC is employed: initial placement is done using the *hierarchical coverage areas* proposal and migrations are executed with the proposed mobile awareness for coverage area change according to the *follow-me principle*. In the hc-fm case, the second level algorithm that chooses a specific edge host within the mobile edge platform manager is always load balancing (the server consolidation algorithm was also investigated, and the results are almost identical to load balancing and thus not presented here).

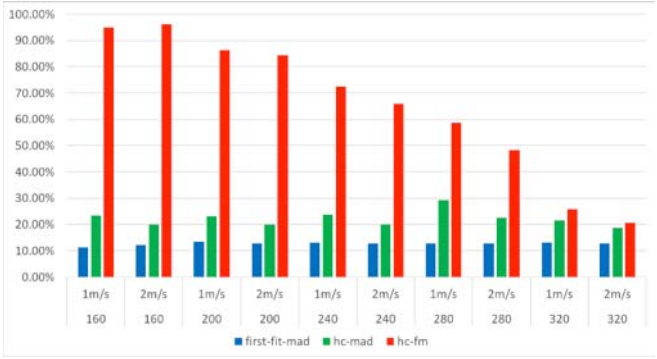
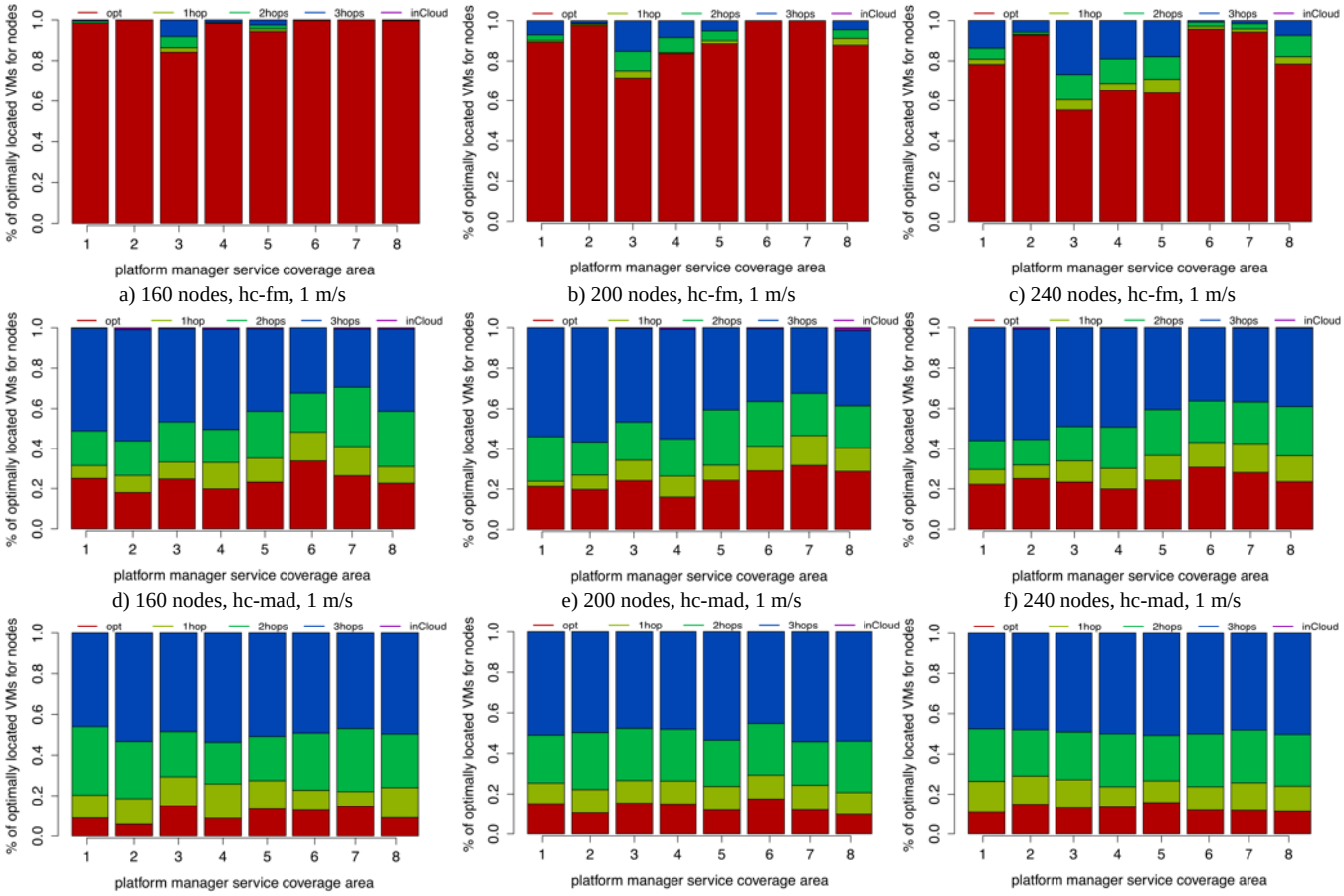


Figure 5. Cross-comparison of the percentage of optimally placed VMs for different number of mobile nodes and average speed combinations

Fig. 5 is a summed-up comparison of the efficiency of the three investigated approaches, where the percentage of optimal mobile node-to-VM communication is presented for different number of mobile nodes with average speed of both 1 and 2 m/s. This summary head-to-head comparison clearly shows the advantages of using the proposed location and mobility-aware approach that dominates even when the resources of the available mobile edge hosts are fully consumed (320 nodes). The figure also provides insight into the influence of the mobile node speed, namely, the stress caused by the increased demand for migrations resulting from faster moving nodes can be felt in the cases of very high loads consuming above 75% of the available resources. In order to make an in-depth analysis of the performance, we also investigated the efficiency of the approaches per coverage area. The results are presented on Fig. 6.

Fig. 6 provides insight into the efficiency of the three scenarios (first-fit-mad, hc-mad and hc-fm) in terms of overall optimality in providing the best possible placement of the serving VM for each mobile node and migrating the VM according to the node mobility pattern. By cross comparison of the results for the same scenario (column-wise) it is evident that the traditional approach of first-fit provides very inefficient results with an average of 50% of nodes being furthest away from the serving VMs (3 hops). In other words, using first-fit provides optimal usage of the edge layer and minimum latency for only 10% of nodes on average. Similar to it, the hc-mad approach, provides optimality only at the initial placement that is lost as the node starts to move, ending up with 20% of nodes being optimally served. It is interesting to note that the performances of these two approaches do not depend on the number of mobile nodes and the load in the fog infrastructure, which is an effect of the migrations policy, that is very rarely used. Also, the effect of the approach is



a) 160 nodes, hc-fm, 1 m/s b) 200 nodes, hc-fm, 1 m/s c) 240 nodes, hc-fm, 1 m/s

d) 160 nodes, hc-mad, 1 m/s e) 200 nodes, hc-mad, 1 m/s f) 240 nodes, hc-mad, 1 m/s

g) 160 nodes, first-fit-mad, 1 m/s h) 200 nodes, first-fit-mad , 1 m/s i) 240 nodes, first-fit-mad , 1 m/s

Figure 6. Efficiency in placement and migration of serving VMs for different algorithms and scenarios

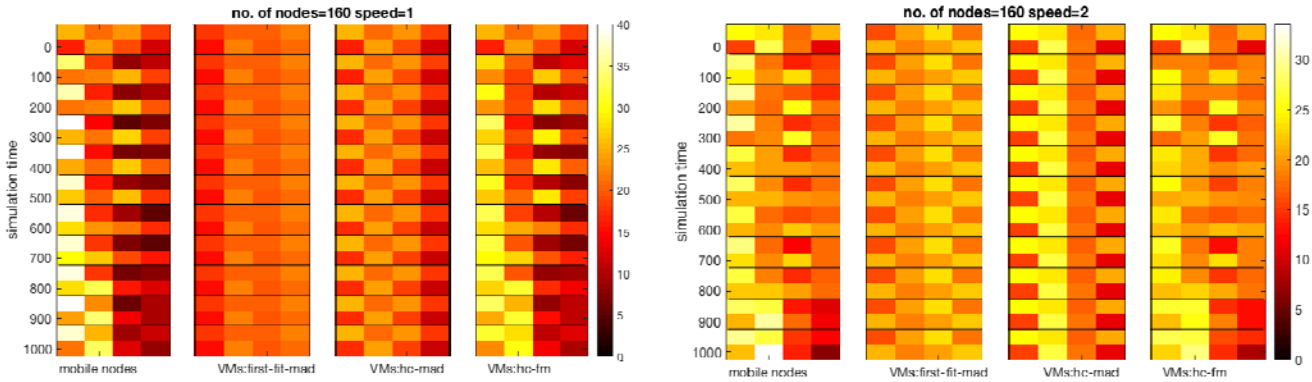
even across the different service areas governed by the mobile edge platforms. The coverage area has no impact on the first-fit performance, while when using hc-mad the difference in performance per coverage area is mainly due to the initial placement process, where it can be seen that the first half of the terrain underperforms compared to the second one. When analysing the mobility patterns of the node, see Fig. 7, this difference can be correlated with the node density and mobility being focused in the first half to center of the terrain.

In comparison, using our proposal, hc-fm, the percentage of optimally serving VMs are ranging from over 90% for 160 nodes, down to average 80% for 240 nodes, Fig. 6.a-6.c. In this case, the efficiency of hc-fm is clearly dependent on the load in the edge layer due to the much larger number of migrations that need to be made in order to ensure the optimal mobile node - VM communication. With the number of VMs rising, there are less available resources to migrate the VMs. When analysing the performances per coverage area, it is evident that some areas are more efficient than others. This effect is due to the choice of mapping the cells to the mobile edge platform managers. The results presented here are using the first mapping option, and thus a small change in the node position may trigger a migration event that requires the VM to be moved to the furthest part of the network, 3 hierarchical hops away the current position. If there are no resources available in the destination platform, the failed migration will result in high latency penalty. In addition, the RandomWayPoint mobility model is such that the node density is higher in the center compared to the edges of the terrain, thus making the central areas more frequently visited and the mapped central mobile edge platform managers resources spread thin. This is why, as the number of nodes grows, the central mobile edge platforms are the first ones to exhibit lower optimality percentages. This effect is presented and analysed in more details in the rest of the results that follow.

C. Follow-me efficiency

The main analysis of interest is the follow-me efficiency, that is the optimality of the initial placement and the subsequent migration events generated each time a node moves from one service area to another one requiring mobile edge platform manager handoff. In Fig. 7 the efficiency of the follow-me implementation is presented by comparing the density of the mobile node distribution on the terrain, and the corresponding VM distribution in the mobile edge platforms for the three different approaches. The *y* axis of the figures is the simulation time, where for each time stamp the density of nodes or VMs in each coverage area of the terrain is presented using a heatmap. The first column of the figure is the mobile nodes heatmap, while the other three correspond to the VMs heatmap when the first-fit-mad, hc-mad and hc-fm approaches are used. The results are presented for node speed of 1 and 2 m/s.

The heatmaps clearly show the efficiency of the follow-me behaviour implemented with hc-fm. In an ideal scenario, the VM density heatmap should be identical to the node density heatmap representing the equal number of VMs and nodes per coverage area leading to optimum minimum latency in node-VM communication. When comparing the node heatmap to the first-fit-mad heatmap, it is clear that



a) node speed 1 m/s b) node speed 1 m/s

Figure 7. Change of node and VM density per terrain coverage area in time. (for each time stamp the 8 coverage areas of the terrain are represented as a 2 x 4 matrix, where the heat corresponds to number of nodes, or VMs location based on the resource management approach used)

this approach makes no attempt to follow the nodes in any way. The first-fit-mad heatmap is almost constant throughout the whole simulation, and while the node density becomes higher in the left part of the terrain, the number of serving VMs in the corresponding platform managers infrastructure is lowest due to the initial node/VM distribution. The hc-mad approach is showing similar problems with the exception of timestamp 0. Since this approach is aware of the node location during VM placement, it is evident that the density node/VM maps at the beginning of the simulation are identical, but the approach fails to adapt to the node mobility. Our proposal, hc-fm starts out the same, but continues to follow the nodes, which is clearly visible with the VM heatmaps in each time stamp that very closely follow the changes in the node heatmap. When comparing the results obtained for different node speeds, there is no change in the follow-me efficiency as the node speed increases.

In the case of node speed of 2 m/s, see Fig. 7.b, there are examples where the node density changes rapidly from one timestamp to another, but the hc-fm VM density is successful in following by increasing the number of necessary migrations. One of these examples can be observed in timestamp 0 and timestamp 100, where the node density in the second row, first column area of the terrain doubles (going from red to yellow) and the change corresponds in the terrain area for the hc-fm VMs location.

D. Investigating different coverage area to mobile edge platform manager mapping

As already mentioned, all of the results presented previously in Fig. 5-7 are for scenarios that are using the Mapping 1 setup, see Fig. 4.a. Using this type of mapping whenever a mobile node moves from one side of the rectangular terrain to another, the change of coverage area results in the need to migrate between two points that are furthest apart in the hierarchical edge network. This type of mapping can be very undesirable in cases when there are migration failures that can occur because of the significant increase network traffic in the MEC layer. In order to investigate how much the mapping from the coverage area to the mobile edge platform manager (that governs a set of edge hosts) can influence the performance of the proposed VM resource management solution, another set of simulation was run using a different type of mapping: Mapping 2, see Fig. 4.b. Fig. 8 represents the total from-to flux of nodes and VMs for each coverage area pair for a scenario of 160 nodes that move with the speed of 2 m/s. The main diagonal equals zero since we are only interested in the events when the nodes change location and move to a different coverage area. The total number of these events averages to about 1100 for the represented scenario.

The initial cross-comparison results showed that the hc-fm approach achieves over 90% in average optimal follow-me efficiency throughout the simulation. However, the per coverage area performances also showed that some areas are more sensitive than others, and their performances are the first to decline when the load in the network increases. It was already mentioned that this is an effect of the chosen *coverage area to platform manager mapping*, and the results presented in Fig. 8 shed more light on this influence.

Fig. 8.a shows that the highest flux in this scenario is from area 1 to area 3, which explains why area 3 is the most sensitive area and shows the lowest results in the per area performance analysis presented in Fig. 8.c. It is also fairly easy to see that area 1 and area 3 have the highest flux with area 1 having the highest flux for nodes leaving the area, while area 3 has the highest flux for nodes entering the area. Since a large number of nodes are coming to the area, there are no enough resources to accommodate all migrations request leading to poor performances. The situation is similar for areas 4, 5 and 8 that also show higher incoming fluxes that leads to faster resource exhaustion.

The efficiency is somewhat better for the second mapping, Fig. 8.b, where now the most sensitive coverage areas are changed so that area 1 becomes the weakest link being the origin for most of the major fluxes to the rest of the coverage areas. Similar situation, but with less frequency, is present for area 2, as well. When comparing the distribution of the flux, mapping 2 provides a more uniform but not necessarily more efficient approach since the effect of the high flux in area 1 is reflected in almost all areas that are part of the terrain. While in both cases there is one significantly active area (3 vs 1), the uniform flux distribution for mapping 2 creates a ripple effect of increased migration requests in all areas all over the terrain putting a higher demand on the resources.

For comparison on the impact these differences in the fluxes for the two mappings have on the overall macro results, in Fig. 9 the percentage of nodes that are on different hop distance from their corresponding VMs in each time stamp are represented for the same case scenario. The

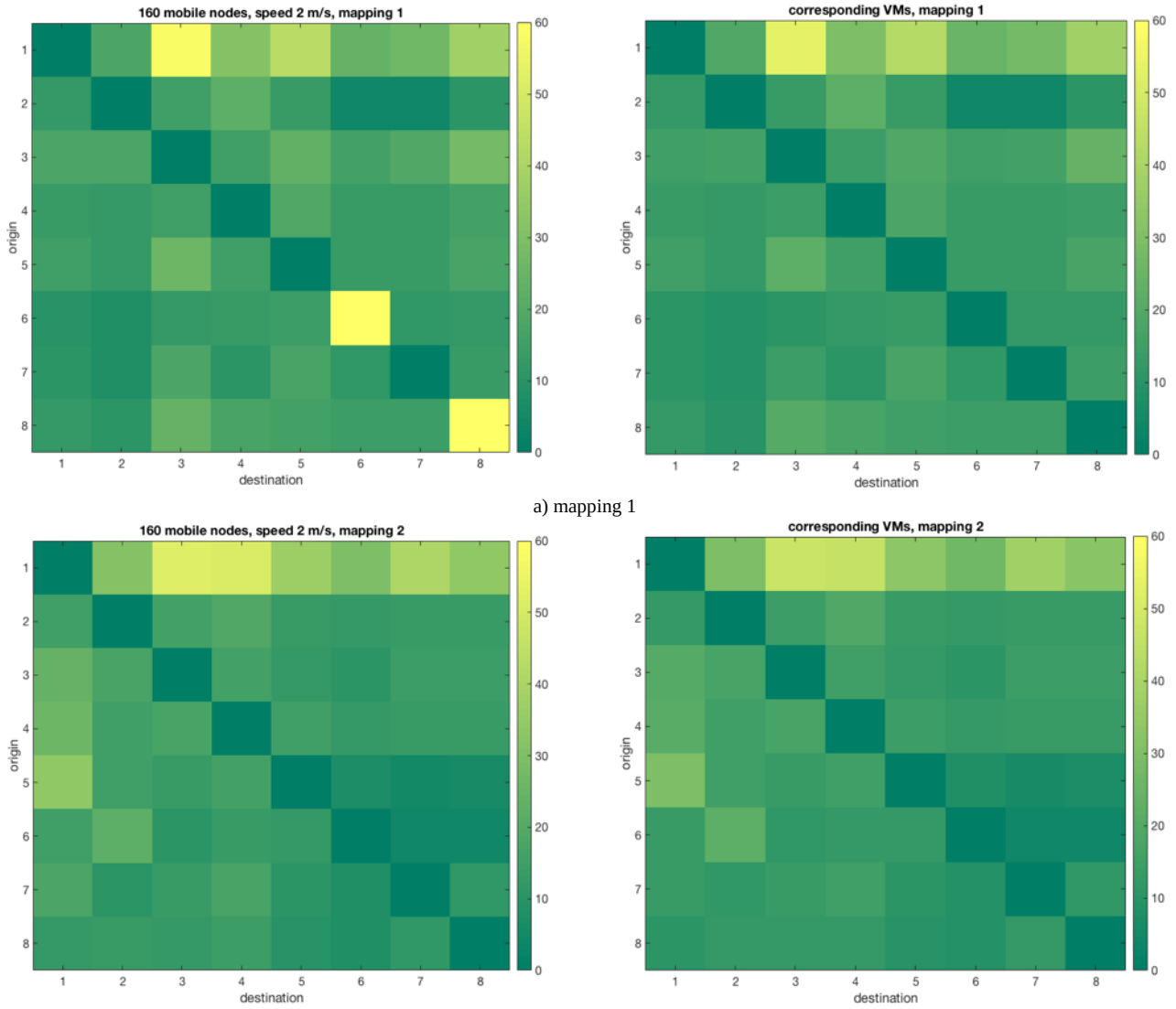


Figure 8. Mobile nodes and serving VMs fluxes from-to coverage area, scenario: 160 nodes, speed 2 m/s.

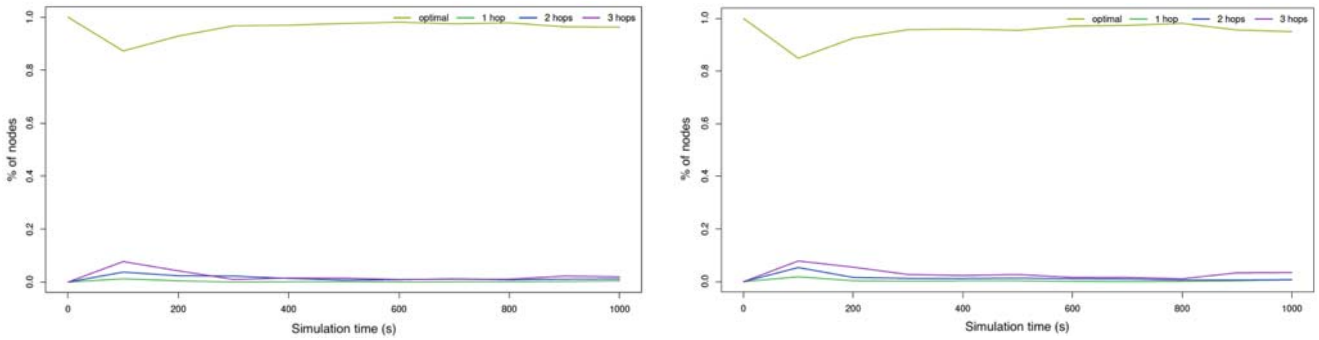
results show that the difference in mapping is minimal on the macro scale, with mapping 2 showing slightly worse results as the simulation time progresses leading to fewer nodes that are 1 hop away from their VMs, and a slightly higher number of nodes that are 3 hops away from their VMs. This is due to the events that occur every time a node moves from the centre of the terrain towards one of its sides that are logically mapped to edge host servers belonging to a distant part of the provider network. In conclusion, the mapping has an effect on the performances of hc-fm on a per coverage area basis, and its careful design can slightly impact the overall performances.

V. CONCLUSION

The main goal of the paper is to analyse the problems of efficient resource management as a part of the mobile edge architecture implementation, with the aim of ensuring the continuously provisioning of low latency via location and mobility-awareness respective to the end user. To efficiently cope with these problems, the responsibilities of the mobile edge orchestrator and mobile edge platform manager where analysed, and the implementation characteristics of the location tracking module, QoS monitoring module and the VM resource management module have been defined.

Within the VM resource management module, as a part of the mobile edge orchestrator, we propose using a hierarchical coverage area approach for optimally packing the individual VMs that will provide the edge services to the mobile nodes during initial placement. The main goal of the approach is to locate the VM in the infrastructure managed by the mobile edge platform manager that is mapped to the coverage area wherein the requesting node is located. Additionally, in order to maintain the low latency node - VMs communication in a mobile environment, a continuous oversight of the handoff events occurring due to the node mobility is implemented using the location tracking and QoS monitoring components. These two components can then help to implement a follow me behaviour of the VM using a specifically defined migration approach. This mobility-aware live migration management attempts to migrate a VM every time a mobile node moves from one coverage area to another, thus moving the VM to mobile edge hosts under the new corresponding platform manager.

The presented results confirm that using the proposed approach the system can achieve a state of continuous efficient follow me behaviour. The cross comparison with traditional approaches to placement and migration shows that the proposed hc-fm solution provides significant



a) mapping 1 b) mapping 2

Figure 9. Percentage of nodes that are at 1 hop, 2 hops and 3 hops away from their serving VMs, scenario: 160 nodes, speed 2 m/s

improvements in the achieved optimality. Using the proposed hierarchical coverage area technique, the complete resource management lifecycle can be managed in an optimal manner even under heavier load conditions. By making an in-depth analysis on the level of each coverage area, the results show that the average efficiency of up to 95% optimal node-VM communications is achieved with different success in different coverage areas, which is a result of the defined mapping between the area and the mobile edge platform manager. The analysis of the node and VM density per coverage area over time shows that the follow-me behaviour is consistently highly efficient in every simulation step. The intensity of the from-to coverage area flux comparison for different mappings provides an example where careful mapping can provide more uniform usage of the resources throughout the total coverage area.

REFERENCES

[1] X. Foukas, G. Patounas, A. Elmokashfi, M. K. Marina, "Network slicing in 5G: Survey and challenges," IEEE Communications Magazine, vol. 55(5), May 2017, pp. 94-100. https://doi.org/10.1109/MCOM.2017.1600951

[2] R. Mahmud, R. Kotagiri, R. Buyya, "Fog Computing: A Taxonomy, Survey and Future Directions," In: Di Martino B., Li KC., Yang L., Esposito A. (eds) Internet of Everything. Internet of Things (Technology, Communications and Computing). Springer, Singapore, 2018. https://doi.org/10.1007/978-981-10-5861-5_5

[3] L. Gao, T. H. Luan, B. Liu, W. Zhou, S. Yu, "Fog computing and its applications in 5G," 5G Mobile Communications, Springer, Oct. 2017, pp. 571-593. https://doi.org/10.1007/978-3-319-34208-5_21

[4] P. H. Kuo, A. Mourad, C. Lu, M. Berg, S. Duquennoy, Y. Y. Chen, C. Y. Li, "An integrated edge and Fog system for future communication networks," IEEE Wireless Communications and Networking Conference Workshops (WCNCW), Barcelona, Spain, April 2018, pp. 338-343. https://doi.org/10.1109/WCNCW.2018.8369023

[5] O. Osanaiye, S. Chen, Z. Yan, R. Lu, K.-K. R. Choo, M. Dlodlo, "From cloud to fog computing: A review and a conceptual live VM migration framework," IEEE Access, vol. 5, pp. 8284-8300, April 2017. https://doi.org/10.1109/ACCESS.2017.2692960

[6] V. Bahl, Cloud 2020: Emergence of micro data centers (cloudlets) for latency sensitive computing (keynote), Middleware April 2015.

[7] A. Manzalini, R. Minerva, F. Callegati, W. Cerroni, A. Campi, "Clouds of virtual machines in edge networks," IEEE Communications Magazine vol. 51 no. 7 pp. 63–70, July 2013. https://doi.org/10.1109/MCOM.2013.6553679

[8] S. Clayman, E. Maini, A. Galis, A. Manzalini, N. Mazzocca, "The dynamic placement of virtual network functions," IEEE Network Operations and Management Symposium (NOMS), Krakow, Poland, May 2014. https://doi.org/10.1109/NOMS.2014.6838412

[9] L. F. Bittencourt, M. M. Lopes, I. Petri, O. F. Rana, "Towards virtual machine migration in fog computing," 10th IEEE International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, Krakow, Poland, Nov. 2015. https://doi.org/10.1109/3PGCIC.2015.85

[10] S. Ningning, G. Chao, A. Xingshuo, Z. Qiang, "Fog computing dynamic load balancing mechanism based on graph repartitioning," China Communications vol. 13 issue 3, pp. 156–164, March 2016. https://doi.org/10.1109/CC.2016.7445510

[11] V. B. C. Souza, W. Ramírez, X. Masip-Bruin, E. Marín-Tordera, G. Ren, G. Tashakor, "Handling service allocation in combined fog-cloud scenarios," IEEE International Conference on Communications, Malaysia, May 2016. https://doi.org/10.1109/ICC.2016.7511465

[12] A. Machen, S. Wang, K. K. Leung, B. J. Ko, T. Salonidis, "Live service migration in mobile edge clouds," IEEE Wireless Communications, Feb. 2018, pp. 140-147. doi:10.1109/MWC.2017.1700011

[13] H. Gupta, A. Vahid Dastjerdi, S. K. Ghosh, R. Buyya, "ifogsim: A toolkit for modeling and simulation of resource management techniques in the internet of things, edge and fog computing environments," Software: Practice and Experience vol. 47 no. 9, 2017, pp. 1275–1296. https://doi.org/10.1002/spe.2509

[14] M. Aazam, E.-N. Huh, "Dynamic resource provisioning through fog micro datacenter", IEEE International Conference on Pervasive Computing and Communication Workshops, USA, March 2015. https://doi.org/10.1109/PERCOMW.2015.7134002

[15] C. T. Do, N. H. Tran, C. Pham, M. G. R. Alam, J. H. Son, C. S. Hong, "A proximal algorithm for joint resource allocation and minimizing carbon footprint in geo-distributed fog computing," 2015 IEEE International Conference on Information Networking (ICOIN), Jan. 2015, pp. 324–329. https://doi.org/10.1109/ICOIN.2015.7057905

[16] S. Wang, J. Xu, N. Zhang, Y. Liu, "A Survey on service migration in mobile edge computing," IEEE Access, vol. 6, 2018, pp. 23511-23528. doi: 10.1109/ACCESS.2018.2828102

[17] K. Velasquez, D. P. Abreu, M. R. M. Assis, C. Senna, D. F. Aranha, L. F. Bittencourt, N. Laranjeiro, M. Curado, M. Vieira, E. Monteiro, E. Madeira, "Fog orchestration for the Internet of Everything: state-of-the-art and research challenges," Journal of Internet Services and Applications 9:14, 2018. https://doi.org/10.1186/s13174-018-0086-3

[18] C. Pahl, B. Lee, "Containers and clusters for edge cloud architectures–a technology review," 3rd IEEE International Conference on Future Internet of Things and Cloud (FiCloud), Rome, Italy, Aug. 2015. https://doi.org/10.1109/FiCloud.2015.35

[19] M. Abo-Zahhad, N. Sabor, S. Sasaki, S. M. Ahmed, "A centralized immune-voronoi deployment algorithm for coverage maximization and energy conservation in mobile wireless sensor networks," Information Fusion vol. 30 issue C, July 2016, pp. 36–51. https://doi.org/10.1016/j.inffus.2015.11.005

[20] E. G. Coffman Jr, J. Csirik, G. Galambos, S. Martello, D. Vigo, "Bin packing approximation algorithms: survey and classification," Handbook of Combinatorial Optimization, Springer NY, 2013, pp. 455–531. https://doi.org/10.1007/978-1-4419-7997-1_35

[21] S. Filiposka, A. Mishev, K. Gilly, "Community-based allocation and migration strategies for fog computing," IEEE Wireless Communications and Networking Conference, WCNC, Barcelona, Spain, April 2018. https://doi.org/10.1109/WCNC.2018.8377095

[22] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, R. Buyya, "Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," Software: Practice and experience 41 (1) (2011) 23–50. https://doi.org/10.1002/spe.995

[23] M. Mishra, A. Sahoo, "On theory of vm placement: Anomalies in existing methodologies and their mitigation using a novel vector based approach," IEEE International Conference on Cloud Computing, Washington, USA, July 2011. https://doi.org/10.1109/CLOUD.2011.38

[24] A. Beloglazov, R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers," Concurrency and Computation: Practice and Experience vol. 24 issue 13, October 2011, pp. 1397–1420. https://doi.org/10.1002/cpe.1867