Mobile Edge Computing for Content Distribution and Mobility Support in Smart Cities

Pedro F. do Prado, Maycon L. M. Peixoto, Marcelo C. Araújo, Eduardo S. Gama, Diogo M. Gonçalves, Matteus V. S. Silva, Roger Immich, Edmundo R. M. Madeira, and Luiz F. Bittencourt

Abstract The pervasiveness of mobile devices is a common phenomenon nowadays, and with the emergence of the Internet of Things (IoT), an increasing number of connected devices are being deployed. In Smart Cities, data collection, processing, and distribution play critical roles in everyday quality of life and city planning and development. The use of Cloud computing to support massive amounts of data generated and consumed in Smart Cities has some limitations, such as increased latency and substantial network traffic, hampering support for a variety of applications that need low response times. In this chapter, we introduce and discuss aspects of distributed multi-tiered Mobile Edge Computing (MEC) architectures, which offer data storage and processing capabilities closer to data sources and data consumers, taking into account how mobility impacts the management of such infrastructure. The main goal is to address topics on how such infrastructure can be used to support content distribution *from and to* mobile users, how to optimize the resource allocation in such infrastructure, as well as how an intelligent layer can be added to the MEC/Fog infrastructure. Furthermore, a multifaceted literature review is given, as well as the open issues and challenging aspects of resource and application management will also be discussed in this chapter.

P. F. do Prado · M. C. Araújo · E. S. Gama · D. M. Gonçalves · M. V. S. Silva · E. R. M. Madeira
L. F. Bittencourt (✉)
Institute of Computing (IC), University of Campinas (UNICAMP), Campinas, Brazil
e-mail: pfprado@unicamp.br; marcelo.araujo@ic.unicamp.br; eduardogama@lrc.ic.unicamp.br; diogomg@lrc.ic.unicamp.br; edmundo@ic.unicamp.br; bit@ic.unicamp.br

M. L. M. Peixoto
Departamento de Ciência da Computação, Federal University of Bahia (UFBA), Salvador, Brazil
e-mail: maycon.leone@ufba.br

R. Immich
Metropolis Digital Institute (IMD), Federal University of Rio Grande do Norte (UFRN), Natal, Brazil
e-mail: roger@imd.ufrn.br

A. Mukherjee et al. (eds.), *Mobile Edge Computing*,
https://doi.org/10.1007/978-3-030-69893-5_19

Keywords Mobile edge computing · Multi-access edge computing · Fog computing · Mobility · Content distribution · Resource allocation · Smart cities

1 Introduction

The evolution of wireless communication networks has changed our interaction as a ubiquitously connected society. This was driven by the number of mobile devices, their ever-increasing hardware capabilities, and systematic cost reductions. To put this into perspective, mobile devices are nowadays prevalent and present an annual growth rate of around 25%. Literature reports such an increase to reach the expected amount of 80 billion mobile devices by 2030 [13]. On top of that, the number of bandwidth-hungry applications is also gaining apace, with the estimated global monthly mobile data traffic expected to raise 3.7 exabytes per month in 2015 to 30.6 exabytes in 2020 [15]. Moreover, these new devices are also expanding their ability to produce data. This leads to a broad collection of information ranging from weather-related data to social behavior, which can be stored, transferred, processed, and analyzed in several distinct ways. This new dynamic reflects on how the devices use the available networks, putting forward stringent resource demands.

At the same time, it is important to notice that the network transformations are continually evolving. In a related manner, the diffusion of the Internet of Things (IoT) will have a central role in this renewal [38]. This technology envisaged that, in essence, all objects would have some type of communications capabilities. This will lead to an unprecedented amount of data that will flood the access networks daily. The integration of both mobile and IoT devices with the Cloud allows alleviating some of these stringent requirements as it provides resource elasticity on-demand, reduces compatibility issues, and provides high availability [16]. However, in doing that, it also introduces new entanglements such as higher latency and core network surcharge as well as security and privacy concerns.

To improve on the aforementioned challenges while increasing the location awareness, Fog and Edge computing can be used. The main idea of both is to provide Cloud-like features (e.g., resource elasticity and virtualization) closer to the end-user. To put in another way, they aim to bring a snippet of the processing power from the Cloud to where the data source and/or devices are [42]. It is worth noticing that this does not mean relinquishing Cloud structures but instead putting it together with Fog and Edge technologies to enable a multi-tier computing hierarchy [7]. This arrangement yields a number of advantages, for example, reduced delays and network traffic as the data can be stored and processed closely [6], which is imperative for delay-sensitive applications. Security and privacy may also be impacted as, in this case, only summarized can be transferred to the Cloud.

The convergence of Cloud, Fog, and Edge computing provides several benefits; on the other hand, it also imposes brand-new constraints and challenges [7]. For example, this architecture needs to be able to handle heterogeneous devices with distinct communication capabilities, uneven processing power, and limited energy-capabilities. Incidentally, the advent of the fifth and sixth generation of wireless systems (5G and 6G) will help furnish this resource-demanding upsurge and better accommodate both network and device heterogeneity. This builds an ecosystem of technologies and value chains aiming to cater to the swift and flexible deployment of innovative services and applications. The 5G systems are designed to provide high bandwidth capacity, low latency, support for dense networks, and improved seamless mobility. In order to enable these highly-desired features, 5G will heavily depend on Mobile (or Multi-access) Edge Computing (MEC), which is standardized by the European Telecommunications Standards Institute (ETSI) and was formally known as Mobile Edge Computing. This adjustment is an attempt to adopt a broader posture regarding which network access technologies will be sanctioned under the proposed framework [7, 29]. This paved a new direction on accepting a comprehensive set of wired and wireless communication technologies and not only carrier-grade cellular equipment.

It is expected that MEC will play a pivotal role in 5G systems by addressing a range of use cases. In order to do that, it aims to bring together the telecommunication-capabilities and the Cloud service environment within the radio access networks (RAN), in the close vicinity to the end-users, and being able to attend applications on a localized basis [50]. Moreover, it can cost-effectively enable high-performance computing on-demand to support a growing number of services and applications at the network's edge. To do that, it will be able to host compute-intensive applications/services and process large chunks of data before sending it to the Cloud. This leads to low latency connectivity and also the possibility to deploy localized content caching.

This chapter brings an overview of problems that have to be addressed to achieve efficient content distribution when mobility is expected to play an important role in the resource management of distributed infrastructures. In Sect. 2, a general view of a multi-tiered Edge computing infrastructure is presented, and the ETSI reference architecture is briefly presented to match requirements. A literature review is presented in Sect. 3. Additionally, Sect. 4 provides details about content distribution and mobility in a MEC scenario. After that, the open challenges are described in Sect. 5, while Sect. 6 brings remarks and concludes the chapter.

2 Multi-Tiered Architecture: Concepts and Definitions

This section introduces the concepts and definitions of MEC and its variations. First, a general view of multi-tiered computing infrastructure for Edge computing and IoT in Smart Cities is presented. Then, it is discussed how this infrastructure can be managed using current standardization efforts.

2.1 Edge and Fog Computing in Smart Cities

Nowadays, Cloud computing has been established as the computing infrastructure to provide computing services to many applications. More recently, Edge and Fog computing [7] are being developed to, in conjunction with the Cloud, improve computing capabilities to fulfill application demands with stricter delay requirements as well as to reduce network traffic by distributing computing capacity closer to the users.

Figure 1 illustrates a scenario where users connect to their access points while traveling in smart cities. Those access points provide cloudlets (fog nodes, or microdata centers) as a first-mile distributed computing capacity, providing lower response times and reducing network traffic to the Cloud by aggregating data and/or fulfilling application computing needs at the Edge. These fog nodes can be arranged in a hierarchy, forming a multi-tiered distributed computing infrastructure from the edge to the cloud.

As users move, for example, in a smart city, their computation should, ideally, be kept as close as possible, i.e., at the cloudlet available in the access point the user is currently connected. Therefore, to manage applications and data from mobile users, management entities distributed in this hierarchy must act to optimize the overall system performance (e.g., response times, utilization, cost, energy consumption).

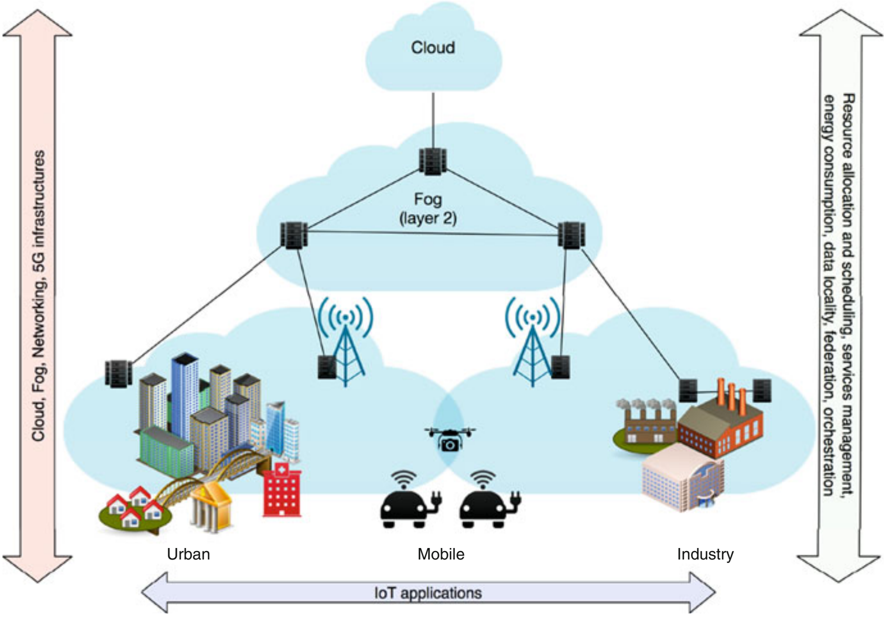
Cloud
Fog
(layer 2)
Cloud, Fog, Networking, 5G infrastructures
Resource allocation and scheduling, services management, energy consumption, data locality, federation, orchestration
Urban
Mobile
Industry
IoT applications

Fig. 1 Overview of a multi-tiered edge computing infrastructure (from [7])

Management and resource allocation in the Edge computing distributed infrastructure brings many challenges, and standardization and specification efforts are under development. One of these efforts is discussed in the next section.

2.2 Mobile Edge Computing Specification

The ETSI Mobile Edge Computing Industry Specification Group (MEC ISG) published a reference architecture for Mobile Edge Computing [18]. The reference architecture is divided into three layers: System Layer, Host Layer, and Network Layer, as illustrated in Fig. 2.

The groups of reference points are divided into (Mp), related to MEC platform functions, (Mm), linked to management; and (Mx), working as external elements connections. The (Mp) group includes ($Mp1, Mp2, Mp3$) reference points. The Mp1 reference point connects the MEC platform to Applications, providing registration and discovery services. The Mp2 reference point manages applications routing between the MEC platform and the Virtualization Infrastructure's Data Plane. The Mp3 controls the communication between MEC platforms. The (Mm) group includes ($Mm1, Mm2, \ldots, Mm9$) reference points. Mm1 is used to instantiate

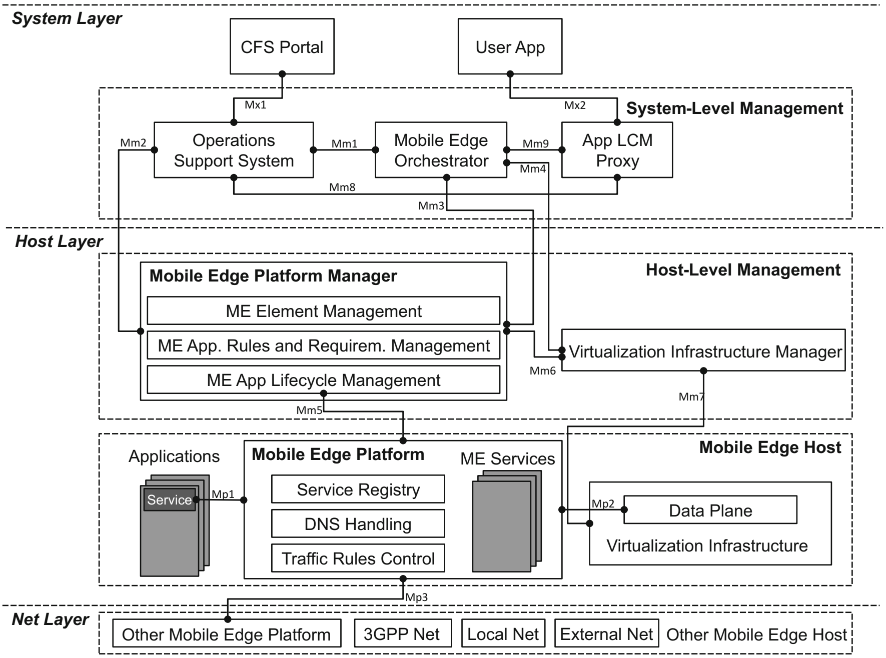

Fig. 2 Mobile edge computing reference architecture [18]

and terminate MEC applications between Operations Support System (OSS) and Orchestrator in the System Layer. Mm2 is responsible for the configuration and performance management between OSS and MEC Platform Manager in the Host Layer. Mm3 manages the lifecycle, application rules, and requirements service between Orchestrator and MEC Platform Manager. Mm4 connects Orchestrator to the Virtualization Infrastructure Manager, and it is used to manage the virtualized resources. Mm5 is for the configuration of applications and services between MEC Platform Manager and MEC Platform in the Host Layer. Mm6 manages the virtualized resources related to the application lifecycle, which is linking the MEC Platform Manager and Virtualization Infrastructure Manager. Mm7 is used to manage the virtualization infrastructure between Virtualization Infrastructure Manager from Host-Level Management and Virtualization Infrastructure from MEC Host. Mm8 connects Orchestrator to App LCM Proxy, and it handles the requests for running applications in the System Layer. Mm9 links Orchestrator to App LCM Proxy, and it is used for MEC application management. The (Mx) group includes $(Mx1, Mx2)$ reference points. Mx1 connects OSS to CFS Portal and deals with third-parties' requests for running applications in the System Layer. Mx2 connects APP LCM Proxy to User App and is used by a device application to request and run an application in the System Layer.

System Layer The upmost layer is the System Layer, composed of Customer-Facing Service (CFS)/Applications and the System-Level Management, which is necessary to run mobile edge applications within an operator network, thus providing system-wide management functions.

The User Application is a mobile edge application running an application requested by a user in the mobile edge system, and the User Application Lifecycle Management Proxy (App LCM Proxy) is the component that deals with the instantiation and termination of the applications. The Customer-Facing Service Portal (CFS) is the first step for providing applications. CFS handles the operations with third-party customers, providing information for instantiation of a set of mobile edge applications that meet specific needs and the termination of these MEC applications. An Mx1 reference point is used to connect CFS to the OSS. OSS manages the operators' network services, which receives and decides on granting requests from the CFS portal and ME Applications. The granted requests are forwarded to the Mobile Edge Orchestrator (MEO) for further processing. MEO has the System Layer's primary function due to wide visibility over the entire mobile network's resources and functionalities. MEO is responsible for maintaining information of all available applications and following their requirements to perform the deploying into mobile edge host [11, 18, 48].

Host Layer At the Host Layer, the Mobile Edge Platform Manager, Mobile Edge Platform, Mobile Edge Host, and the Virtualization Infrastructure are used to execute the user applications.

Mobile Edge Platform Manager (MEPM) is an entity that is further divided into Mobile Edge Element Management, Mobile Edge Application Rules, Requirements Management functions, and Mobile Edge Application Lifecycle Management. Mm3 reference point connecting the MEPM to MEO provides support for the application and services in the System Layer. Mm2 reference point linking MEPM and OSS is used for fault reports, configuration, and performance measurements received from the Virtual Infrastructure Manager via Mm6 reference point. Meanwhile, VIM is responsible for allocating, managing, and releasing the virtualized resources, such as compute, storage, and network, to the mobile edge applications [18, 48].

The Mobile Edge Platform (MEP) is responsible for offering services such as discovering and advertising to the mobile edge applications. MEP is also used to manage the networking environment by handling the service registry, DNS configuration, and the traffic rules control accordingly [18].

The Virtualization Infrastructure is located in or close to the network edge, e.g., the Network Functions Virtualization Infrastructure (NFVI), which offers virtualized resources to mobile edge applications. Moreover, the virtualization infrastructure brings a Data Plane that runs traffic rules from MEP and manages the traffic among services, applications, DNS, 3GPP, and other local and external networks [18].

Network Layer The Network Layer is further related to the connectivity to cellular networks (3GPP), Local and External networks such as the Internet. The Host Layer consists of Mobile Edge Host and the Host-Level Management. However, to include the benefits of heterogeneous access technologies to the MEC, e.g., 4G, 5G, and WiFi, ETSI ISG changed the name of Mobile Edge Computing (MEC) to Multi-access Edge Computing in 2017 [29], maintaining the acronym MEC. In this chapter, we use the general term MEC to refer to this architecture's latest developments. From this expansion, Fig. 3, the intelligence is moved to the, bringing communication functionalities as well as computation, caching, and additional control services. The overall layering organization remains similar to the previous one, but the network layer has been modified to consider multiple different access technologies.

The integration of MEC and 5G is shown in Fig. 3. In addition to Radio Access Network (RAN) and User Equipment (UE), the main 3GPP 5G network functions are briefly summarized below.

- User Plane Function (UPF): controls the plane operations and may even be part of the MEC Layer in some specific deployments.
- Authentication Server Function (AUSF): acts as an authentication server.
- Session Management Function (SMF): performs the session management functions.
- Access and Mobility Management Function (AMF): handle the procedures related to mobility and deals with the RAN control plane.
- Network Slice Selection Function (NSSF): selects the network slice resources and AMF for users.
- Network Repository Function (NRF): maintains the network functions and their supported services.
- Unified Data Management (UDM): deals with users and subscription services.

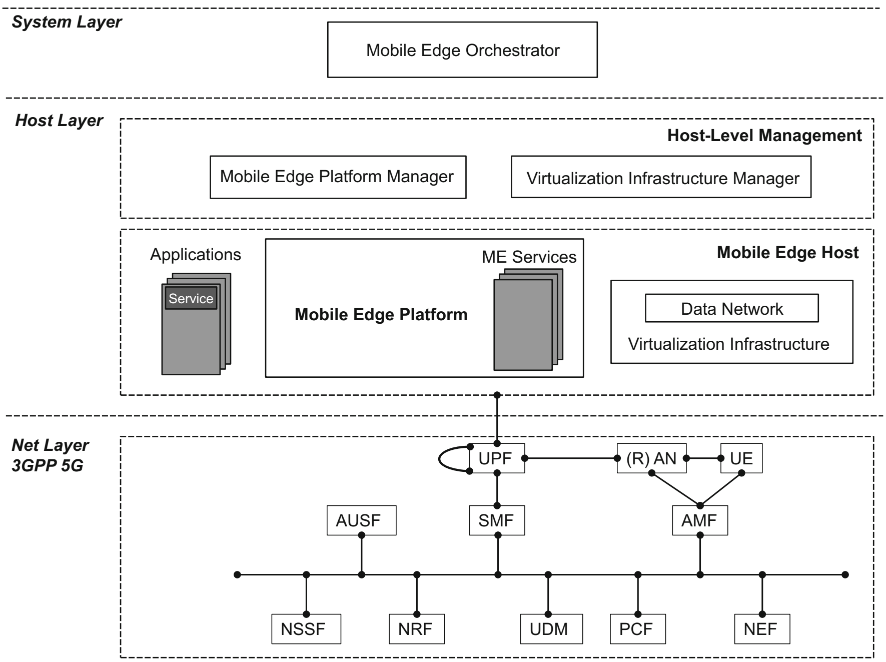

Fig. 3 MEC and 5G architecture [29]

- Policy Control Function (PCF): handles network policies and rules in the 5G control plane.
- Network Exposure Function (NEF): acts as a service that manages all access requests outside the system.

The ETSI reference architecture brings an overview of the management needs to support application mobility at the network's edge. Several algorithms and mechanisms need to be incorporated into the architecture to provide reduced delays and improved Quality of Service (QoS). The remainder of this chapter discusses a few problems that, when addressed, can provide better support for mobile applications in smart city scenarios.

3 Literature Review

MEC's main idea is to offer processing and storage services at the Edge of the network, increasing computing services proximity to users. Recently, MEC architecture is a topic that has been gaining attention from industry and academic researchers: several surveys analyze the state of the art, discuss definitions, and identify the main challenges to be overcome. In [59], Wang et al. surveyed caching, communication, and computing issues at the Edge of the network. Mach et al. [36] surveyed existing MEC concepts, functionalities, mobility awareness, and computing offloading. In [21], Habibi et al. surveyed architectural distinctions between existing Edge computing models and analyzes the different aspects of the practical implementation of Fog computing, such as security, computing resource management, networks, and systems design. Furthermore, Abbas et al. [2] surveyed architectures, application areas, and highlights futures directions related to MEC.

Although Edge and Cloud infrastructure composition is a topic that has been extensively investigated, mobility management is one of the biggest challenges to be yet overcome in MEC. For example, in [27] the authors analyze the impact of mobility on the caching process at the Edge of the network. In specific scenarios, the characteristics and properties of users' mobility are unknown. Some research [35, 39] proposed strategies to predict user's mobility. The information obtained through prediction allows the content management process's actions to offer a higher Quality of Experience (QoE) for users.

Due to the importance of ensuring continuity of access to content and services during users' movement, migration also emerges as one of the core research issues in the context of MEC. In [32], a decision policy is proposed to determine when the VM migration process should start—after each handoff performed by the user, a decision is made based on the trade-off between the gain and cost of migration. The authors modeled the decision policy using the Continuous-Time Markov Decision Process (CTMDP). Moreover, the Follow-me Cloud [51] concept proposes that users' content should be migrated, on-demand, to the cloudlet closest to the user, reducing latency and improving the QoS offered.

Mobility management in MEC can impact several types of applications, among which video delivery is a trendy one. MEC architectures for video delivery offers an environment characterized by high bandwidth and low latency. In [25, 47, 54, 62], the authors focus on decreasing the traffic overload in the network core and improving the QoE aided by MEC. Yang et al. [62] explore machine learning models to incorporate into the MEC node for decision-making on storing popular videos. Experimental results suggest good performance for mobile video streaming services. Furthermore, Petrangeli et al. [41] proposed an advanced architecture in which additional intelligent components are placed to support video delivery. Instead of considering low-level network performance parameters, the designed network components focus on optimizing the QoE parameters that directly affect users' experience.

Rectal and Benkacem et al. [5, 46] propose a content delivery network as a service (CDNaaS), where content providers can create a CDN slice that includes cache, transcoder, and streamers for several videos for their users. The objective is to find an efficient cost for creating a slicing following requirements of the network administrator in terms of QoE and the cost of setting up the Cloud infrastructure.

MEC enables data collected at the Edge to be processed at the Edge. Associating the high amount of data from IoT with the MEC architecture [66], it is possible to explore new applications and services at the Edge when data collection is allied with Artificial Intelligence (AI). Large data sets generated at the Edge of the network along with benefits brought by MEC urges for distributed intelligence to be supported close to the end-users.

In [60, 66], the authors explore the feasibility of Deep Learning (DL) in terms of applications and how to improve networking aspects to make it possible to deploy DL at the Edge. From this perspective, the work of [33] refers to the use of Federated Learning (FL), a method that executes DL at the Edge of the network using distributed local user data, requiring the transmission of only the learning model in the aggregation period.

Valério et al. [55] focuses on energy usage by choosing to go a layer upwards and do more work in Fog, distributing the learning through the cloudlets. This way, it is possible to have energy gains using short-range technologies with little loss of precision. Their work discusses that the type of wireless technology cannot directly impact intelligence, but how energy and traffic must be well aligned with the chosen wireless technology.

Park et al. [40] also considers wireless networks, but with a focus on modeling methods for both learning and its algorithms to fit the principle of providing the most learning at the most extreme point possible. Zhang et al. [65] follow this line of learning to model but explores MEC in vehicles. Offloading and edge caching is essential for good management of aspects of the network, with the use of storage resources and extra resources.

This book chapter aims to congregate the discussion on how mobility management in MEC can impact the applications and the Edge-Cloud infrastructure.

4 Content Distribution and Mobility

Significative growth in mobile connectivity is expected in the next few years. The addition of mobile users will undeniably change the dynamics of MEC environments. Introducing mobility support in a multi-tier MEC translates some traditional resource management problems, such as service placement and routing path calculation, into a more complex and dynamic case. Aiming to deal with such a scenario, the MEC infrastructure requires new approaches to orchestrate this environment. For example, the adoption of static or dynamic service allocation and content migration can result in distinct levels of QoE delivered to the users and different resource usage in the MEC infrastructure.

Figure 4 illustrates this situation where the user's latency is affected when the application is static or dynamically allocated. It is important to notice that if the offloaded data/processing is migrated along with the user in his/her path, the application delays can be kept at lower levels. In a scenario where there is no migration (illustrated by the blue line), as the user moves away from the cloudlet where his/her content is allocated, the delay increases, degrading the QoS. On the other hand, the red line represents the scenario where content is constantly migrated to the cloudlet that is closest to the user at a given time, keeping latencies as low as

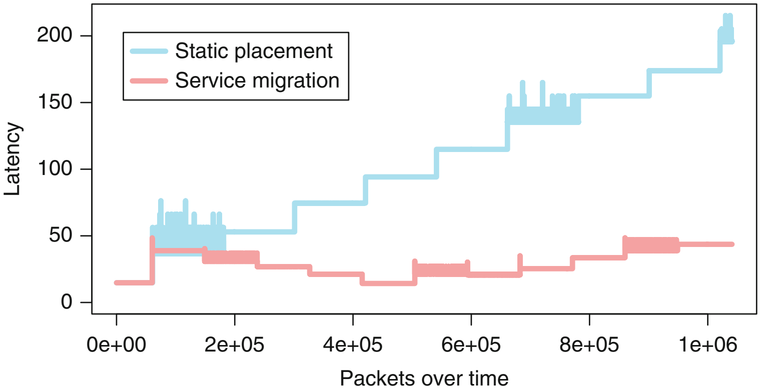

Fig. 4 Latency provided by the Fog in scenarios with and without VM migrations

possibly supported by the Fog architecture. The interested reader can find detailed results and different mobility scenarios in [44].

In such a dynamic environment, the MEC architecture needs to deal with user mobility to deliver the required content. In this scenario, the content distribution needs to take advantage of MEC architecture not only for caching data but also using MEC's processing capacities to, for instance, perform real-time video transcoding in a faster way, also to avoid data transfer into the core network towards the Cloud. Such dynamism in the density of the network increases the complexity of the content distribution problems.

This section discusses the impact of mobility support in the orchestration of MEC infrastructures and the role of content distribution and processing in such an environment. Additionally, it is shown several typical management and optimization problems related to data collection, distribution, and processing in scenarios with user mobility. Several uses cases are going to be addressed, such as the migration of mobile users data/applications throughout the multi-tiered infrastructure, the support for high-definition video streaming for mobile users, and the use of machine learning in an intelligent edge layer for vehicular traffic and safety.

4.1 Mobility and Content Migration

Similar to the Cloud paradigm, MEC can provide its resources in the form of virtualized environments, such as containers or virtual machines (VMs), providing an isolated environment that contains all the resources required by the user.

The constant movement of devices is one of the biggest challenges for MEC architecture as it needs to be able to reduce the latency between the user and his/her content. According to Yan et al. [61], the study of human mobility shows that people tend to visit specific places at constant time intervals, setting standards.

In the literature, it is possible to find several mobility models that identify human movement characteristics in different scenarios [17, 58].

The study of mobility is important to identify movement patterns, allowing the process of content migration (e.g., VM or container migration) to occur according to each user's movement's particularities. Mobility models also provide the possibility of a proactive migration approach; that is, the content can be migrated in advance to locations that the user is likely to move to [19].

To guarantee the quality of the mobile user experience, it is necessary to decrease the physical distance between content and users. In the literature, there are several strategies [31, 51, 64] for content management at the network's edge. In general, they propose user's content should be dynamically allocated according to their current position. In such scenarios, whenever a mobile user changes him/her position, the relevant application/contents should be moved from a host server to another one in closer proximity to the current user position.

Recently, proactive content management strategies have been gaining attention from the scientific community. Proactive strategies [4, 19] aim to predict when and where users will need their content/applications in order to perform management decisions efficiently, ensuring the quality of the users' experience and highly improving the QoS for delay-sensitive applications.

In this context, maintaining the application (geographically or logically) as close as possible to the user is a great challenge [43], mainly due to a trade-off in the migration process. Frequent migrations will allow greater proximity between user and content, resulting in lower latency. However, migrating too often between cloudlets may enlarge the application downtime, which is not desirable. Furthermore, in scenarios with a large number of users, frequent migrations will congest the network, compromising its stability and reducing the QoS offered by the infrastructure. On the other hand, insufficient migrations may keep applications away from their users, resulting in increased latency. Both scenarios can impair mobile users' QoS. Therefore, MEC architecture's content management strategies must offer solutions to improve user experience quality without compromising the network operation.

The problem of content migration at the Edge of the network with mobile users has been a focus of researchers' attention, with several different approaches being proposed at this time. Several metrics and criteria can be used to define when and whether the user's content should be migrated, such as latency and throughput requirements, load balancing, user speed, or application priority [44]. Further discussion on this can be found in the following section. Once the mobile users change their locations, the MEC node serving them may not fulfill their application requirements (e.g., latency) anymore. Other objectives, such as load balancing or energy-saving, can also lead to migrations in the architecture.

Figure 5 depicts a hierarchical MEC architecture. A number of different combinations of origin and destination nodes can be found in the migration process. The user content can be moved in both horizontal and vertical directions in terms of the MEC nodes' hierarchical organizations. Horizontal migrations occur between MEC nodes at the same hierarchical level, as illustrated as the type 1 migration. Once the

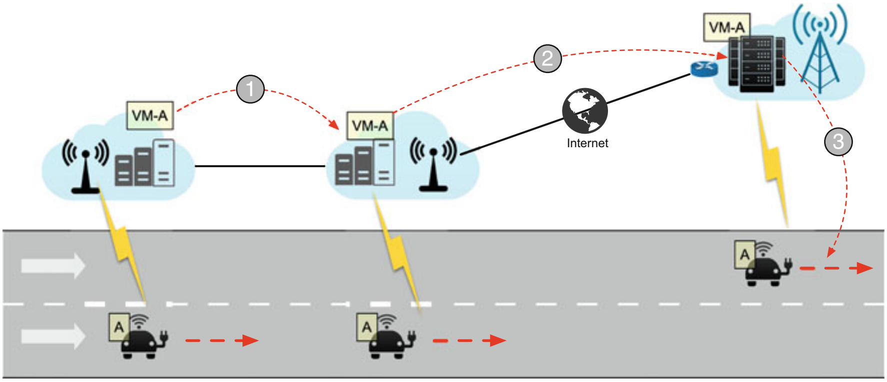

Fig. 5 VM migration scenarios in a fog computing architecture

MEC nodes, at the same level, usually have similar computing power, that kind of migration tends to keep, in a stable range, the levels of QoS offered to the users. Furthermore, vertical migrations may also be needed if, for example, the users' requirements or the MEC's resources availability change dynamically over time. If the user increases their requests for computing power, for instance, the resources demand on the current MEC node may surpass its capability. A resource richer MEC node, used to be closer to the network's core, should fulfill the new user's requirement (illustrated as the type 2 migration). Similarly, if the user decreases him/her computing and storage requirements or the required latency should be improved, the user content can be migrated to a MEC node closer to the Edge of the network (type 3 migration). Moreover, fog nodes at higher levels can be used to reduce the number of migrations (and, consequently, downtime) when, for example, high-speed users are moving at the Edge.

User access to content can also be performed in different ways based on the user's location and his/her content. Access can be direct if the content is one hop away from the user, which provides the lowest possible latency. However, if necessary, multiple hops may be traversed when user contents are not placed in the closest fog node.

Therefore, one main concern of models like MEC is the proximity between mobile users and their contents, as well as reducing network congestion while maintaining QoS for mobile users [14]. However, the development of strategies to realize this management is far from trivial. In the light of that, it is necessary to develop solutions that orchestrate content migration considering several aspects, such as user mobility patterns, characteristics of applications, migration costs, networking utilization and congestion, and so on.

4.2 Resource Allocation and Optimization

One of the main questions that arise in the MEC architecture is how to distribute heterogeneous services and their data throughout the MEC hierarchy such that application requirements are obeyed, even in a constrained infrastructure. For example, latency is one of the utmost importance requirements for many interactive edge applications. Scheduler decision making heavily relies on such requirements to model an optimization problem that outputs the resource allocation that determines where applications and their components should run.

Furthermore, the possibility of mobile users continuously requesting resources from the MEC infrastructure has a considerable effect on the environment which needs to be managed. In a static scenario, once the resources were allocated to a user, it will be reserved until the tasks are finished. In this case, changes in the resources demand will only occur when the number of active users increases or decreases. On the other hand, in the mobility scenario, this change of resource demand in the infrastructure also happens when a relevant number of users move to a specific area. Figure 6 illustrates a scenario where a large number of mobile users can be temporally concentrated in a reduced part of the map. The red circle

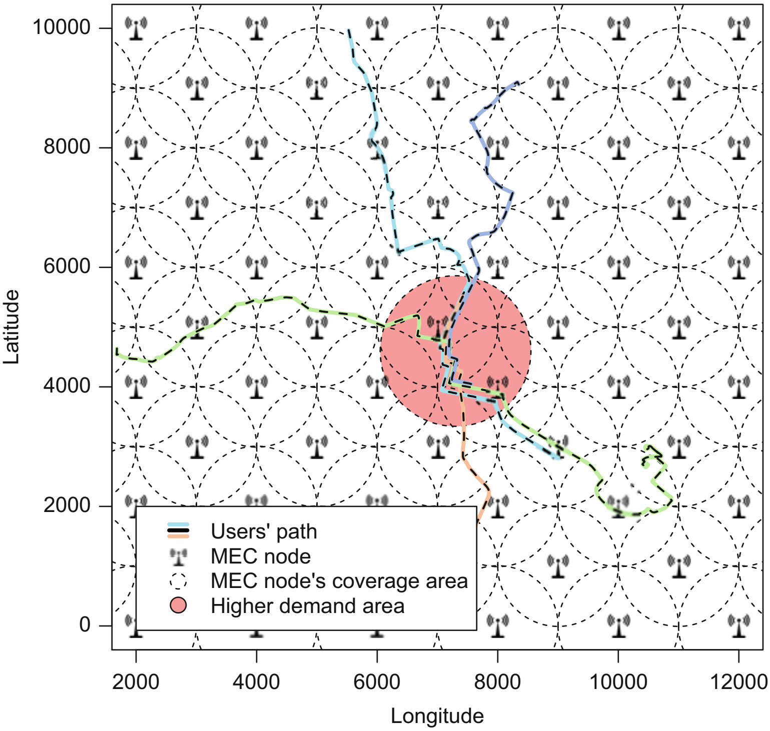

Fig. 6 Mobile user converging to a common area resulting in a resource concurrency

illustrates such an area. In this case, MEC allocation approaches need to deal with such a dynamic environment. Circles represent base-stations/access points antenna range, where each of these access points has a cloudlet offered to the mobile users. Illustrative routes from these users are illustrated in blue, purple, orange, and green. When mobile users reach a certain position, which can be pre-defined or calculated in real-time, a decision-making algorithm should run to choose the best new location (cloudlet) for his/her VM, based on current mobility (e.g., prediction of next position) and also on current load of cloudlets in the user's path. Based on that, good resource management approaches are a goal for both users, which then experience a better QoE, and MEC resource providers, which can improve profit metrics as energy or network usage by better allocating resources.

The resource allocation imposes challenging problems in distributed systems. In other to alleviate these issues, one prominent component of the MEC infrastructure is the scheduler. This component is responsible for allocating resources from the distributed infrastructure to users' applications. The decision-making on which applications should run where is taken by following an optimization model. In general, it takes information about the application requirements and infrastructure characteristics as input. The scheduler defines one or more objective functions, whose output maximizes or minimizes single or multiple objectives.

Considering that, one important aspect of MEC architectures is related to users' mobility along the infrastructure edge. In MEC, optimization modeling should consider user mobility as a determining factor of future allocation needs. How to model this problem in a hierarchical infrastructure will impact which optimization technique is more appropriate: when a larger scope is considered, faster optimization is needed. When mobility is added, the environment's dynamics brings the need for faster optimization techniques to be developed for this computing model. The scheduler should then run an optimization algorithm to find the best allocation possible for applications considering requirements and resources capacity.

In general, the scheduling problem, which involves finding the optimal solution among a universe of exponential possibilities, is considered a hard problem to solve among computer problems. Most of these problems are classified into classes of problems named as NP-Complete and NP-Hard [28]. One of the most classical problems is known as Knapsack Problem (KP) [30]. This problem consists of, based on a set of items with different sizes and a knapsack with a defined capacity, finding the optimal set of items that maximize the use of the knapsack capacity.

To exemplify the KP in the MEC architecture context, consider a system with five cloudlets and 100 VMs. The basic version of the scheduler just needs to find a valid solution for the allocation of the VMs in the cloudlets, considering the needs of each VM (e.g., processing power, memory) and the capacities of each cloudlet. This can be expanded to consider multiple cloudlets at the same level, multiple levels, and also include the Cloud. As users move, the problem becomes dynamic in nature, and the modeling should be adapted to be able to find solutions in a reasonable time. Different optimization algorithms and techniques can be applied to behave according to the current dynamicity observed in the system.

Many different types of algorithms can be used to solve optimization problems, such as the classic KP problem and its variations. The most basic is an exhaustive search (or Brute force) that will test for each possible valid solution and find the best one. In practice, this algorithm can be used only in very small search space sizes because it has an exponential execution time. However, if the search space size is small enough, it always guarantees the global optima (best solution). In the VM placement problem for MEC, the search space can vary in size: from a single cloudlet to the whole hierarchy. Brute force may be a choice for local optimization in a single cloudlet when a few users are currently at that location.

Another class of algorithms used to solve optimization problems, including the KP problem, is Dynamic Programming. The Branch and Bound algorithm focus on solving combinatorial optimization problems. Basically, in combinatorial optimization, the choices to be made are discrete (i.e., where to allocate each VM), and in continuous optimization, the choices to be made are based on continuous values (i.e., real numbers). Dynamic programming can speedup the scheduler solution to be applicable in larger scenarios, e.g., considering multiple fog nodes and hierarchical levels.

Other optimization techniques include heuristics and meta-heuristics, where solution quality is not guaranteed, but the algorithms running times are reduced. For example, artificial intelligence has also been used to solve KP problems. Some examples are Genetic Algorithms (GA), Ant Colony Optimization (ACO), as well as hybridizations combining two or more techniques. Heuristics and meta-heuristics are suitable for larger and more dynamic scenarios, where multiple runs of the optimization are needed to keep the objective functions optimized. This is clearly the case in MEC, for example, in rush hours when a great percentage of users move around and need to have their VMs/containers properly placed to improve QoS and obey requirements.

4.3 Streaming Services

Combining Edge and Cloud computing environments bring to streaming services attractive improvements in terms of bandwidth usage and reduced latency. End-users can expect high-quality video applications to work anywhere and on a variety of heterogeneous devices, including mobile ones. In Video-on-Demand (VoD) services, the edge resources of Internet Service Providers can be utilized to host video contents in the proximity of end-users, thereby reducing latency and mitigating load on core networks and data centers. This is especially helpful for live streaming scenarios that require low latency [24, 47]. Moreover, pre-processing can be done in multiple streaming flows deployed at the Edge. Consequently, reducing the download traffic needed from the Cloud. An edge architecture for video streaming delivery has the following purposes: (1) Improving the users QoE, serving the requested edge content as close as possible to the user; (2) Reduce congestion at

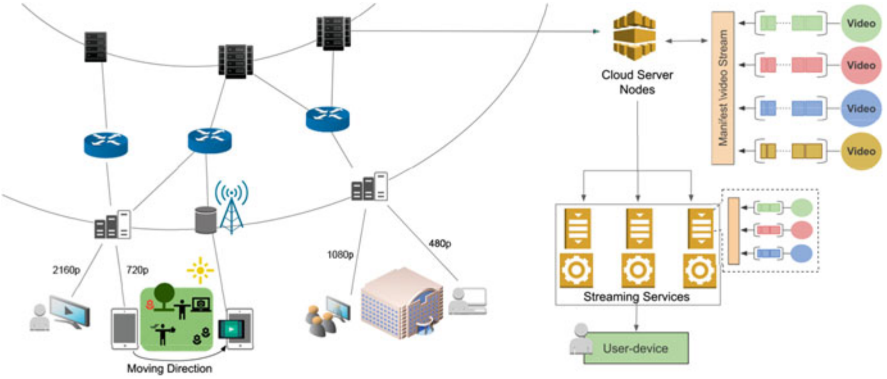

Fig. 7 Illustration of collaborative video delivery on a MEC network

the core of the network; (3) Efficiently deal with the amount of data that needs to be processed and extract meaningful data to create more intelligence.

Figure 7 depicts a network service scenario that uses intelligent video streaming in a MEC architecture. A MEC server (i.e., an fog node) is connected to base stations to perform data storage and processing. A MEC client can access video streaming services being run in the infrastructure [62]. This video service can cache videos and run analytics to extract knowledge about video content and video service performance, such as estimating QoE from throughput for different users. This can assist network-level decisions to adjust the data rate accordingly to the available downlink bandwidth, presenting real-time network information and context in addition to reduced latency.

To provide cache services in a MEC architecture, it is important to effectively deliver the video content through smart caching mechanisms. Such mechanisms can be based, for example, on content popularity and geographical location/distribution of mobile devices. With this strategy, it is possible to efficiently use VoD and live broadcast services to a wide range of heterogeneous devices. In order to improve this, a good idea is to distribute the service closer to the region with more bandwidth consumption. This approach is similar to the existing overlay cache that is applied to services with lower latency indexes due to edge utilization. This can lead to the improvement of the QoE for the majority of users. In other words, smart caching available on the fog nodes enable popular videos to be available closer to the user, thus reducing traffic load and delay [54].

Besides promoting caching, a MEC architecture can also perform data processing at the Edge. Figure 7 gives another example, the deployment of a transcoding service closer to the end-users can improve the QoE in dense networks with heterogeneous resolutions being requested. For instance, transcoding of cached videos can be run in a MEC server when a user requests a different version. This task can be run in the MEC server that stores the original video (data provider node) or the MEC node serving the video (delivery node). For example, a video with a 5 Mbps (720p) bit rate could be transcoded from a cached copy presenting a bit rate of 8 Mbps (1080p). In doing that, the fog node uses the bandwidth available to serve as many users as possible. Moreover, the content provider does not waste bandwidth, sending high bit rate video through the core network.

MEC infrastructures can be utilized to store and process video closer to the user, performing real-time transcoding, caching for reduced bandwidth use, video analytics, augmented reality, and so on.

4.4 Intelligence at the Edge

Edge devices produce large amounts of data nowadays, enabling the so-called Smart Environments, also as a consequence of the current pervasiveness of personal mobile and IoT devices [56]. The massive data source has considerably changed in this scenario, moving from Cloud data centers to end devices. Bringing Artificial Intelligence (AI) and Machine Learning (ML) to be run at the Edge of the network is seen as a possibility to enable the full potential of Big Data processing in MEC infrastructures.

In the standard Big Data scenario, data is generated at the Edge of the network and must be transported to data centers, which contains a very high processing and storage capacity. Then, AI is applied to generate knowledge about those data and keep it in a central location. Data centers are often geographically far from end-users, which implies in transferring a large volume of data across links, resulting in increased latency and congestion. In MEC, AI can be applied at the Edge as well, processing local data to generate knowledge about specific regions, but can also aggregate and send data to the Cloud for additional processing to generalize the knowledge with a wider view from the data gathered at the Edge.

Machine Learning provides the most prominent set of tools currently to achieve the mentioned AI objectives, to gain insights, perform classifications and predictions through training with data obtained at the Edge in a process with feed-forward and backpropagation [49]. Among ML methods, Deep Learning (DL) stands out for its unique performance in many tasks. DL is a variation of Neural Networks (NN), which can then be called Deep Neural Networks (DNN). DNNs can learn high-level resources by providing highly accurate inferences on tasks. As shown in Fig. 8, DL works with several neurons in the entrance, called Input Layer, which receives raw data. It is connected to middle layers, known as the hidden layer, that they are going to perform complex operations of learning, sending their results to the output layer. The hidden layer gives more complexity than a Simple Neural Network, which requires more computational power; however, it gives better work results in learning tasks.

Run the DNN models on edge devices requires large computing capacity for DNN algorithms. Therefore, actual intelligence at the Edge depends on architectures and mechanisms able to maintain accuracy by running learning algorithms collaboratively at the Edge in a distributed way, and, complementary, using the synergy

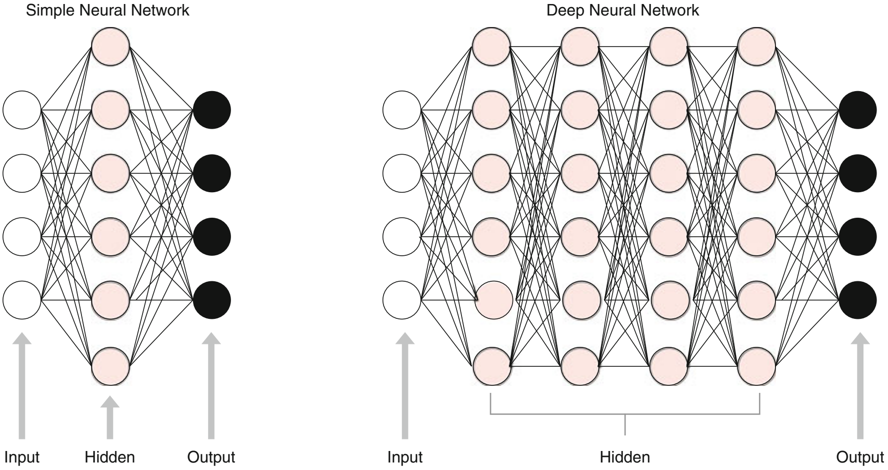

Fig. 8 Deep learning structure

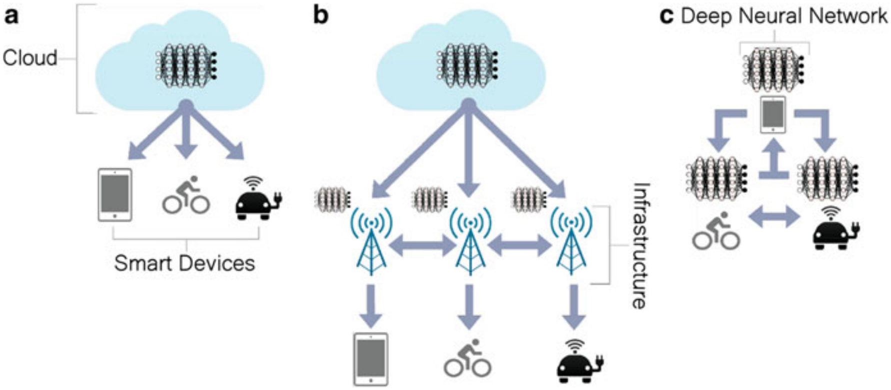

Fig. 9 Three architectures for DNN training: (**a**) Cloud to devices (**b**) Devices keep training DNN models and (**c**) Cloud to Edge Infrastructure, then, to devices

between the Cloud and the Edge in the considered MEC computing hierarchy. In this scenario, the AI model's training can be carried out in the Cloud data centers, which then makes the trained model available to the edge devices at synchronization rounds. Also, Edge computing can use data center resources when necessary to optimize DNN training since transmitting DNN models through the network is less expensive than transferring all raw data to the Cloud.

According to Zhou et al. [66], there are three ways to architect DNN training: centralized, decentralized, and hybrid (Fig. 9):

(a) Centralized: the most common, carried out in data centers;
(b) Decentralized: aims to train models directly on the edge devices, updating the models from time to time;

(c) Hybrid: combines the two above, training of DNN models in data centers and making them available at the Edge.

The first (a) is the classic one, carried out in data centers. The second (b) aims to train the models directly on the edge devices, updating the models from time to time. Finally, the hybrid (c) that combines the two with the training of DNN models in data centers and making them available at the Edge.

Items *b* and *c* involve edge mobile devices running learning models, but *c* still uses the Cloud infrastructure as a form of support. The learning model is massively trained in the Cloud and then forwarded to the Edge infrastructure and then to the devices. Here, end devices will not have the task of training the model but only applying it in accordance with the current application. Adjustments and updates to the model are made with the help of the Edge infrastructure.

Item *b*, on the other hand, covers the entire process of constituting the model, from training the model to its applicability, going through model updates, without the support of other infrastructures. The details of this process are at the discretion of the chosen harvesting method. What matters in this context is that it is all done on mobile devices.

Computationally, learning methods require a lot of resources, so applying them to mobile devices is not straightforward. To avoid significant changes in the architecture of mobile devices (memory, processing, and storage capacity), the way forward would be to optimize the models in order to make them as light as possible [10].

An interesting method is compressing the models [45]. This method generalizes a learning structure by removing weights or operations that are less useful for predictions and divides a large model into smaller models, each focused on a specific application scenario. This improves processing performance as fewer operations are done to achieve similar results.

Some care with the use of computational resources must be taken into account. Bonawitz et al. [8] makes some recommendations in this regard. A new function when learning on the mobile device is the use of a local data repository for training and model evaluation. It is recommended to use simple and small databases like SQLite, and its available storage size is small and non-negotiable, in addition to establishing an expiration date for the data.

As for the user experience and battery life, it is preferred to invoke the learning method only when the device is idle and with a sufficient battery is remaining or when it is connected to the charger. Finally, a cleaning of temporary resources must be scheduled as soon as their execution is complete.

Technologies dedicated to working with distributed DNN at the edge are currently under development, such as Federated Learning [12], Aggregation Frequency Control [22], Gradient Compression [34], DNN Splitting [37], Knowledge Transfer Learning [57], and Gossip Training [9]. Many of these methods work with local data on the device, which increases data privacy.

The availability of high-quality intelligent services is a combination of the chosen AI method with computing performance and network data transfer. Some metrics are adopted to describe the QoS of the Edge computing model inference, such as latency, precision (DNN model), energy efficiency, data privacy, and communication overhead. Introducing mobility with MEC gives rise to new challenges in intelligence at the Edge, where now, more dynamic evaluation of distributed learning models is necessary. For example, in smart cities, traffic management can be performed mostly at the Edge with data from users/vehicles. A more precise estimate of traffic can be performed in real-time with low latency at the Edge, while relevant data are still sent to the Cloud for model training and to produce a wider view of the current traffic landscape. Therefore, in this scenario, the MEC architecture acts to reduce data transmission to the Cloud as well as to produce faster and more precise results in this mobility scenario.

5 Challenges

Since the MEC paradigm is an architecture that extends the Cloud computing concept, it can share some common solutions of other distributed systems. However, the MEC architecture's unique characteristics make it seek new approaches to manage its environment properly. Many approaches have been proposed over the last years, however, many problems still have no definitive solutions. This section introduces some of the open problems present in MEC's development that still are challenging the area.

5.1 Resource Management

In order for users to take advantage of the new possibilities offered by the MEC architecture, storage and computing resources are distributed across the edge of the network, ensuring access to infrastructure for any users who wish to use MEC services. Identifying strategies to define where physical servers with computational resources should be allocated is a significant challenge. To allow resources to be used efficiently, physical servers must be allocated based on users' expected demand, ensuring that users' QoE and QoS requirements are met.

The development of routines capable of managing computational resources is another challenge to ensure MEC's proper functioning. For the resources contained in the MEC infrastructure to be orchestrated efficiently, it is necessary to define signaling messages capable of transmitting information about the status of the resources, such as capacity, availability, and workload. However, signaling messages must not occur so frequently as not to compromise the performance of the MEC infrastructure, and at the same time, they cannot be rare enough so that resource information becomes outdated. The multi-tiered layout in a MEC architecture can help resource management to be performed more efficiently, but proper mechanisms should be designed to work in this computing hierarchy.

5.2 Mobility Management

Besides the requirements of MEC users in terms of computing, storage, and network resources, the MEC infrastructure also needs to manage their different mobility patterns. For example, when moving by foot, bicycle, car, bus, or train, each one of them presenting a specific route and speed. That characteristic of mobile users introduces several challenges for the MEC infrastructure in terms of service availability. Such a dynamic scenario affects different MEC's resource management processes such as load balance, service placement and migration, packet routing, and handoff.

Different researches have been made to increase the capability of MEC infrastructures to support mobile users. However, the impact of user mobility is not completely understood in these infrastructures. More accurate algorithms for predictive mobility patterns can help some processes to plan their future demand in a specific area. Based on such information, the infrastructure can prepare the required resources to serve that increasing demand. This process can prepare MEC to scale up or scale down or even triggers a load balance, service migration, or caching data.

Furthermore, in this context, technologies like Software Defined Networks (SDN), Network Function Virtualization (NFV) [63], and network slicing [3] have been introduced to increase the flexibility of these infrastructures. Besides the capability of network slicing to dynamically reallocate MEC resources to serve these mobile users [20], further studies on that context need to be made to evaluate the computing overhead of that resource reallocation.

5.3 Data Transmission

Although the allocation of computing and storage resources is a key point in the impact of QoE guarantees, the management of network resources can either perform several improvements or impair the user experience. Due to the close relationship between IoT, MEC, and big data, a colossal amount of data is transferred between different architecture points. To properly serve delay-sensitive applications, placing data and computing close to the users is not enough if the MEC architecture cannot deliver the users' requests within the required deadline. Based on that, transmission techniques need to be optimized enough to provide a good connection among the MEC nodes and their users. Moreover, in MEC, it is expected that different wireless and wired technologies work together and seamlessly. Inter-operation and seamless connection maintenance among a variety of protocols is a challenge yet to be overcome.

In this scenario, both wireless technologies and routing protocols must be optimized to provide a faster and more stable connection to the users. One of the main challenges of developing these protocols is dealing with the trade-off between energy efficiency, latency, reliability, and throughput. Predictive offloading [1] and transmission protocols that avoid wireless package collision and improve latency and throughput are some candidate solutions. End-to-end network slicing has also been rising as a promising solution.

5.4 QoS and QoE Guarantees

MEC servers can help guarantee QoS for latency-sensitive applications from mobile users using a resource reservation method. Whereas for static latency-tolerant users, the MEC management system can perform on-demand provisioning to allocate computational resources and provide reliable computing services. However, provisioning schemes that have to take into account high-mobility users is a complex task. Therefore, novel hybrid MEC server schemes must be developed to enable increased MEC providers' revenue through serving a maximum number of users with guarantees on their QoS requirements.

Research studies in QoE show that the changing conditions of best-effort networks introduce numerous problems. In traditional video streaming, each client typically streams a video that is available in a single bitrate on the server-side [23, 26]. A MEC architecture should exploit users' context information to optimize content management and video delivery, which may result in better utilization of network resources and QoE.

5.5 Intelligence at the Edge

Introducing intelligence techniques at the Edge comes with new challenges that must be faced. When considering a MEC learning network, the data can be distributed to be processed on more than one node. In this situation, the development of a tool that offers an automatic and efficient partitioning is both a challenge and an opportunity in this scenario. It is also interesting to note that offloading a training model from the Cloud to the edge nodes can incur high communication costs, especially when considering applications that require persistent training models.

Studying the trade-offs between transferring data to the Cloud and implementing adaptive learning models at the edge, as well as designing adaptation mechanisms for model distribution, is a current challenge. It is important to note that distributed learning also comes with the challenge of privacy-preserving mechanisms, which should also be developed for sensitive data (e.g., medical applications).

Mobility within MEC gives rise to additional challenges for proper implementation of intelligence at the Edge, where a more dynamic evaluation of distributed learning models is necessary. For example, in smart cities, traffic management can be performed mostly at the Edge with real-time data from users/vehicles. A more precise estimate of traffic can be performed in real-time with low latency at the Edge, while relevant data are still uploaded to the Cloud for model training and to produce a wider view of the current traffic landscape. Other mobile application scenarios can present the same characteristics, where a dynamic composition of data from edge devices is crucial for the learning model to provide relevant results.

5.6 Green MEC

Energy consumption has gained attraction from researchers in different areas of computing, such as embedded systems and resource management in Cloud computing and networking. For example, to manage the increase in energy consumption, the InterSCity project[1] has different approaches to this challenge. In a CF-RAN architecture proposed in [52, 53], they introduce local nodes closer to the users where they perform part of the processing tasks. The cloud-level nodes manage the workloads sent to the fog on demand to process the surplus traffic from the front-haul.

The computation on fog nodes is performed through virtualized network functions, where they are activated or deactivated in real-time depending on the network demand. In [55], IoT data are collected at the Edge by nodes called gateways. Communication between The IoT devices and gateways is done using wireless cellular technology. Then, they are used to train a distributed machine learning solution model. Both energy consumption and training performance are evaluated with different configurations and compared to the centralized cloud model. Such a distributed solution significantly mitigates the traffic sent to the Cloud. On the other hand, a reduction in distributed learning precision training has to be made. With the network's edge addition, the energy consumption shows savings of over 90% in data transmission and 2% in precision loss when compared to the centralized cloud-level. Further studies in energy management with mobility and Edge computing are still needed to tackle heterogeneous devices' complexities in a Smart City and mobility.

Each MEC node can use considerably less power than a conventional large Cloud data centers. At the same time, it has lower processing power, requiring a higher number of active locations. Because of that, the increase of new small-scale MEC servers being created becomes a big concern for energy consumption. This way, it is unquestionable to develop innovative techniques for achieving power energy saving. At the same time, computational resources need to be manageable to guarantee satisfactory computational performance.

The small area serviced by each MEC server impacts resource allocation and service management, especially when considering user mobility. The consequence of this architecture is a highly dynamic workload, with a fast change in load patterns. More advanced prediction techniques could be developed to enable optimized resource utilization, focusing on load distribution and reduced power consumption. Moreover, management services for dynamic scaling workloads that require significant computational resources need to be developed. Also, note that

[1] www.interscity.org.

as MEC systems grow over a region, a green load balancing solution needs to be optimized in the best way using further available renewable energy.

6 Conclusion

We have presented in this chapter a distributed, multi-tiered Mobile Edge Computing (MEC) architecture. MEC was introduced by the ETSI Mobile Edge Computing Industry Specification Group (MEC ISG) as a means of offering data storage and processing closer to data sources and data consumers, taking into account the mobility aspects impact on infrastructure management.

We have covered topics on the benefits of using the MEC architecture, such as support for content distribution to mobile users, optimization of resource allocation, video delivery, and intelligence at the Edge. Besides, we have pointed out that MEC was designed to offer low latency connectivity for delay-sensitive applications due to users' proximity at the network's edge.

It becomes clear that many research challenges are still essential to be carried out to properly manage data and resources in MEC architectures, especially with the high heterogeneity in application requirements and into the future. We discussed some directions, providing insights on interesting potential problems for further research.

References

1. Aazam, M., Zeadally, S., Harras, K.A.: Offloading in fog computing for iot: Review, enabling technologies, and research opportunities. Future Generation Computer Systems **87**, 278–289 (2018)
2. Abbas, N., Zhang, Y., Taherkordi, A., Skeie, T.: Mobile edge computing: A survey. IEEE Internet of Things Journal **5**(1), 450–465 (2017)
3. Afolabi, I., Taleb, T., Samdanis, K., Ksentini, A., Flinck, H.: Network Slicing and Softwarization: A Survey on Principles, Enabling Technologies, and Solutions. IEEE Communications Surveys Tutorials **20**(3), 2429–2453 (thirdquarter 2018). https://doi.org/10.1109/COMST.2018.2815638
4. Araújo, M.C., Curado, M., Sousa, B.M., Bittencourt, L.F.: Cmfog: Proactive content migration using Markov chain and madm in fog computing. In: Proceedings of the 13th IEEE/ACM International Conference on Utility and Cloud Computing (2020)
5. Benkacem, I., Taleb, T., Bagaa, M., Flinck, H.: Optimal vnfs placement in cdn slicing over multi-cloud environment. IEEE Journal on Selected Areas in Communications **36**(3), 616–627 (March 2018). https://doi.org/10.1109/JSAC.2018.2815441
6. Bittencourt, L., Diaz-Montes, J., Buyya, R., Rana, O., Parashar, M.: Mobility-aware application scheduling in fog computing. IEEE Cloud Computing **4**(2), 26–35 (March 2017). https://doi.org/10.1109/MCC.2017.27
7. Bittencourt, L., Immich, R., Sakellariou, R., Fonseca, N., Madeira, E., Curado, M., Villas, L., DaSilva, L., Lee, C., Rana, O.: The internet of things, fog and cloud continuum: Integration and challenges. Internet of Things **3–4**, 134 – 155 (2018)

8. Bonawitz, K.A., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., Kiddon, C.M., Konečný, J., Mazzocchi, S., McMahan, B., Overveldt, T.V., Petrou, D., Ramage, D., Roselander, J.: Towards federated learning at scale: System design. In: SysML 2019 (2019), https://arxiv.org/abs/1902.01046, to appear
9. Boyd, S., Ghosh, A., Prabhakar, B., Shah, D.: Randomized gossip algorithms. IEEE transactions on information theory **52**(6), 2508–2530 (2006)
10. Caldas, S., Konečný, J., McMahan, B., Talwalkar, A.: Expanding the reach of federated learning by reducing client resource requirements (2018), https://arxiv.org/abs/1812.07210
11. Carrega, A., Repetto, M., Gouvas, P., Zafeiropoulos, A.: A middleware for mobile edge computing. IEEE Cloud Computing **4**(4), 26–37 (2017)
12. Chen, Q., Zheng, Z., Hu, C., Wang, D., Liu, F.: Data-driven task allocation for multi-task transfer learning on the edge. In: 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS). pp. 1040–1050. IEEE (2019)
13. Chettri, L., Bera, R.: A comprehensive survey on internet of things (iot) toward 5g wireless systems. IEEE Internet of Things Journal **7**(1), 16–32 (2020)
14. Chiang, M., Shi, W.: Nsf workshop report on grand challenges in edge computing. In: Tech. Rep. (2016)
15. Cisco: Cisco visual networking index: Global mobile data traffic forecast update, 2015–2020. Tech. Rep. 1 (2016)
16. Curado, M., Madeira, H., da Cunha, P.R., Cabral, B., Abreu, D.P., Barata, J., Roque, L., Immich, R.: Internet of Things - Next Generation Cyber-Physical Systems, pp. 381–401. Springer (2019)
17. Cuttone, A., Lehmann, S., González, M.C.: Understanding predictability and exploration in human mobility. EPJ Data Science **7**(1), 2 (2018)
18. ETSI, M.: Mobile edge computing (mec); framework and reference architecture. ETSI, DGS MEC **3** (2016)
19. Gonçalves, D., Velasquez, K., Curado, M., Bittencourt, L., Madeira, E.: Proactive virtual machine migration in fog environments. In: 2018 IEEE Symposium on Computers and Communications (ISCC). pp. 00742–00745. IEEE (2018)
20. Gonçalves, D., Puliafito, C., Mingozzi, E., Rana, O., Bittencourt, L., Madeira, E.: Dynamic network slicing in fog computing for mobile users in mobfogsim. In: Proceedings of the 13th IEEE/ACM International Conference on Utility and Cloud Computing (2020)
21. Habibi, P., Farhoudi, M., Kazemian, S., Khorsandi, S., Leon-Garcia, A.: Fog computing: A comprehensive architectural survey. IEEE Access (2020)
22. Hsieh, K., Harlap, A., Vijaykumar, N., Konomis, D., Ganger, G.R., Gibbons, P.B., Mutlu, O.: Gaia: Geo-distributed machine learning approaching {LAN} speeds. In: 14th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 17). pp. 629–647 (2017)
23. Immich, R., Cerqueira, E., Curado, M.: Adaptive qoe-driven video transmission over vehicular ad-hoc networks. In: IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). pp. 227–232 (April 2015). https://doi.org/10.1109/INFCOMW.2015.7179389
24. Immich, R., Cerqueira, E., Curado, M.: Towards a qoe-driven mechanism for improved h.265 video delivery. In: Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net). pp. 1–8 (June 2016). https://doi.org/10.1109/MedHocNet.2016.7528427
25. Immich, R., Villas, L., Bittencourt, L., Madeira, E.: Multi-tier edge-to-cloud architecture for adaptive video delivery. In: 2019 7th International Conference on Future Internet of Things and Cloud (FiCloud). pp. 23–30 (Aug 2019). https://doi.org/10.1109/FiCloud.2019.00012
26. Immich, R., Borges, P., Cerqueira, E., Curado, M.: Adaptive motion-aware fec-based mechanism to ensure video transmission. In: IEEE Symposium on Computers and Communication (ISCC). pp. 1–6 (June 2014). https://doi.org/10.1109/ISCC.2014.6912571
27. Jarray, C., Giovanidis, A.: The effects of mobility on the hit performance of cached d2d networks. In: 2016 14th international symposium on modeling and optimization in mobile, ad hoc, and wireless networks (WiOpt). pp. 1–8. IEEE (2016)

28. Karp, R.M.: Reducibility among combinatorial problems. In: Complexity of computer computations, pp. 85–103. Springer (1972)
29. Kekki, S., Featherstone, W., Fang, Y., Kuure, P., Li, A., Ranjan, A., Purkayastha, D., Jiangping, F., Frydman, D., Verin, G., et al.: Mec in 5g networks. ETSI white paper **28**, 1–28 (2018)
30. Kellerer, H., Pferschy, U., Pisinger, D.: Multidimensional knapsack problems. In: Knapsack problems, pp. 235–283. Springer (2004)
31. Kikuchi, J., Wu, C., Ji, Y., Murase, T.: Mobile edge computing based vm migration for qos improvement. In: 2017 IEEE 6th Global Conference on Consumer Electronics (GCCE). pp. 1–5. IEEE (2017)
32. Ksentini, A., Taleb, T., Chen, M.: A Markov decision process-based service migration procedure for follow me cloud. In: 2014 IEEE International Conference on Communications (ICC). pp. 1350–1354. IEEE (2014)
33. Lim, W.Y.B., Luong, N.C., Hoang, D.T., Jiao, Y., Liang, Y.C., Yang, Q., Niyato, D., Miao, C.: Federated learning in mobile edge networks: A comprehensive survey. arXiv preprint arXiv:1909.11875 (2019)
34. Lin, Y., Han, S., Mao, H., Wang, Y., Dally, W.J.: Deep gradient compression: Reducing the communication bandwidth for distributed training. arXiv preprint arXiv:1712.01887 (2017)
35. Liu, L., Guo, J., Zhang, S., Zhu, J.: Similar user assisted mobility prediction. In: 2019 11th International Conference on Wireless Communications and Signal Processing (WCSP). pp. 1–6. IEEE (2019)
36. Mach, P., Becvar, Z.: Mobile edge computing: A survey on architecture and computation offloading. IEEE Communications Surveys & Tutorials **19**(3), 1628–1656 (2017)
37. Mao, Y., Yi, S., Li, Q., Feng, J., Xu, F., Zhong, S.: A privacy-preserving deep learning approach for face recognition with edge computing. In: Proc. USENIX Workshop Hot Topics Edge Comput.(HotEdge). pp. 1–6 (2018)
38. Mckinsey, Company: Mapping the value beyond the hype. Executive Summary pp. 1 – 144 (2015)
39. Nadembega, A., Hafid, A.S., Brisebois, R.: Mobility prediction model-based service migration procedure for follow me cloud to support qos and qoe. In: 2016 IEEE International Conference on Communications (ICC). pp. 1–6. IEEE (2016)
40. Park, J., Samarakoon, S., Bennis, M., Debbah, M.: Wireless network intelligence at the edge. Proceedings of the IEEE **107**(11), 2204–2239 (2019)
41. Petrangeli, S., Wauters, T., Turck, F.D.: Qoe-centric network-assisted delivery of adaptive video streaming services. In: 2019 IFIP/IEEE Symposium on Integrated Network and Service Management (IM). pp. 683–688 (April 2019)
42. Pisani, F., de Oliveira, F., Gama, E.S., Immich, R., Bittencourt, L.F., Borin, E.: Fog computing on constrained devices: Paving the way for the future iot. Advances in Edge Computing: Massive Parallel Processing and Applications **35**, 22 (2020). https://doi.org/10.3233/APC200003
43. Puliafito, C., Mingozzi, E., Anastasi, G.: Fog computing for the internet of mobile things: Issues and challenges. In: 2017 IEEE International Conference on Smart Computing (SMARTCOMP). pp. 1–6 (2017)
44. Puliafito, C., Gonçalves, D.M., Lopes, M.M., Martins, L.L., Madeira, E., Mingozzi, E., Rana, O., Bittencourt, L.F.: Mobfogsim: Simulation of mobility and migration for fog computing. Simulation Modelling Practice and Theory **101**, 102062 (2020)
45. Ravi, S.: Custom on-device ml models with learn2compress (05 2018), https://ai.googleblog.com/2018/05/custom-on-device-ml-models.html
46. Retal, S., Bagaa, M., Taleb, T., Flinck, H.: Content delivery network slicing: Qoe and cost awareness. In: 2017 IEEE International Conference on Communications (ICC). pp. 1–6 (May 2017)
47. S. Gama, E., Immich, R., F. Bittencourt, L.: Towards a multi-tier fog/cloud architecture for video streaming. In: 2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion). pp. 13–14 (2018)

48. Sabella, D., Vaillant, A., Kuure, P., Rauschenbach, U., Giust, F.: Mobile-edge computing architecture: The role of mec in the internet of things. IEEE Consumer Electronics Magazine **5**(4), 84–91 (2016)
49. Svozil, D., Kvasnicka, V., Pospichal, J.: Introduction to multi-layer feed-forward neural networks. Chemometrics and intelligent laboratory systems **39**(1), 43–62 (1997)
50. Taleb, T., Samdanis, K., Mada, B., Flinck, H., Dutta, S., Sabella, D.: On multi-access edge computing: A survey of the emerging 5g network edge cloud architecture and orchestration. IEEE Communications Surveys Tutorials **19**(3), 1657–1681 (thirdquarter 2017). https://doi.org/10.1109/COMST.2017.2705720
51. Taleb, T., Ksentini, A.: Follow me cloud: interworking federated clouds and distributed mobile networks. IEEE Network **27**(5), 12–19 (2013)
52. Tinini, R.I., Batista, D.M., Figueiredo, G.B.: Energy-efficient vpon formation and wavelength dimensioning in cloud-fog ran over twdm-pon. In: 2018 IEEE Symposium on Computers and Communications (ISCC). pp. 521–526. IEEE (2018)
53. Tinini, R.I., Batista, D.M., Figueiredo, G.B., Tornatore, M., Mukherjee, B.: Low-latency and energy-efficient bbu placement and vpon formation in virtualized cloud-fog ran. IEEE/OSA Journal of Optical Communications and Networking **11**(4), B37–B48 (2019)
54. Tran, T.X., Hajisami, A., Pandey, P., Pompili, D.: Collaborative mobile edge computing in 5g networks: New paradigms, scenarios, and challenges. IEEE Communications Magazine **55**(4), 54–61 (2017)
55. Valerio, L., Conti, M., Passarella, A.: Energy efficient distributed analytics at the edge of the network for iot environments. Pervasive and Mobile Computing **51**, 27–42 (2018)
56. Valerio, L., Passarella, A., Conti, M.: A communication efficient distributed learning framework for smart environments. Pervasive and Mobile Computing **41**, 46–68 (2017)
57. Wang, J., Zhang, J., Bao, W., Zhu, X., Cao, B., Yu, P.S.: Not just privacy: Improving performance of private deep learning in mobile cloud. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 2407–2416 (2018)
58. Wang, M., Yang, S., Sun, Y., Gao, J.: Human mobility prediction from region functions with taxi trajectories. PloS one **12**(11), e0188735 (2017)
59. Wang, S., Zhang, X., Zhang, Y., Wang, L., Yang, J., Wang, W.: A survey on mobile edge networks: Convergence of computing, caching and communications. IEEE Access **5**, 6757–6779 (2017)
60. Wang, X., Han, Y., Leung, V.C., Niyato, D., Yan, X., Chen, X.: Convergence of edge computing and deep learning: A comprehensive survey. IEEE Communications Surveys & Tutorials (2020)
61. Yan, X.Y., Wang, W.X., Gao, Z.Y., Lai, Y.C.: Universal model of individual and population mobility on diverse spatial scales. Nature communications **8**(1), 1639 (2017)
62. Yang, S., Tseng, Y., Huang, C., Lin, W.: Multi-access edge computing enhanced video streaming: Proof-of-concept implementation and prediction/qoe models. IEEE Transactions on Vehicular Technology **68**(2), 1888–1902 (2019)
63. Zaidi, Z., Friderikos, V., Yousaf, Z., Fletcher, S., Dohler, M., Aghvami, H.: Will SDN Be Part of 5G? IEEE Communications Surveys Tutorials **20**(4), 3220–3258 (Fourthquarter 2018). 10.1109/COMST.2018.2836315
64. Zhang, C., Zheng, Z.: Task migration for mobile edge computing using deep reinforcement learning. Future Generation Computer Systems **96**, 111–118 (2019)
65. Zhang, J., Letaief, K.B.: Mobile edge intelligence and computing for the internet of vehicles. Proceedings of the IEEE (2019)
66. Zhou, Z., Chen, X., Li, E., Zeng, L., Luo, K., Zhang, J.: Edge intelligence: Paving the last mile of artificial intelligence with edge computing. Proceedings of the IEEE **107**(8), 1738–1762 (2019)