Cluster Computing
https://doi.org/10.1007/s10586-023-04050-6

Network-assisted processing of advanced IoT applications: challenges and proof-of-concept application

Higinio Mora[1] · Francisco A. Pujol[1] · Tamai Ramírez[1] · Antonio Jimeno-Morenilla[1] · Julian Szymanski[2]

Received: 27 October 2022 / Revised: 8 March 2023 / Accepted: 19 May 2023

Abstract
Recent advances in the area of the Internet of Things shows that devices are usually resource-constrained. To enable advanced applications on these devices, it is necessary to enhance their performance by leveraging external computing resources available in the network. This work presents a study of computational platforms to increase the performance of these devices based on the Mobile Cloud Computing (MCC) paradigm. The main contribution of this paper is to research the advantages and possibilities of architectures with multiple offloading options. To this end, a review of architectures that use a combination of the computing layers in the available infrastructure to perform this paradigm and outsource processing load is presented. In addition, a proof-of-concept application is introduced to demonstrate its realization along all the network layers. The results of the simulations confirm the high flexibility to offload numerous tasks using different layers and the ability to overcome unfavorable scenarios.

Keywords Internet of Things · Mobile computing · Computer networks · Distributed computing · Quality of service · Cloud computing

1 Introduction

The recent advances in computer science and communication technologies have enabled the development of systems that focus on improving productivity and the quality of life in society. Within this context, the information collected from the environment and delivered to users can provide additional value for the development of advanced applications. This trend is encouraging the development of new concepts in many areas of industry and consumer electronics, such as the Internet of Things (IoTs), Cyber–Physical Systems (CPS), and Mobile Computing.

On the other hand, the Cloud Computing paradigm has opened the doors to a dramatic increase in performance for many applications. The US National Institute of Standards and Technology (NIST) provides an accepted conception of this paradigm [1]. Briefly, it is described as a computing technology that takes advantage of cloud resources to enable ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction [2].

The convergence of the IoT and cloud computing paradigms improves their synergy and provides a better user experience [3, 4]. Many new sensors and embedded systems have high computing capabilities that allow them to execute sophisticated applications (e.g., video and image processing, augmented reality, artificial intelligence

✉ Francisco A. Pujol
fpujol@ua.es

Higinio Mora
hmora@ua.es

Tamai Ramírez
tamai.ramirez@ua.es

Antonio Jimeno-Morenilla
jimeno@ua.es

Julian Szymanski
julian.szymanski@eti.pg.edu.pl

[1] Department of Computer Technology and Computation, University of Alicante, Carretera de San Vicente s/n, San Vicente del Raspeig, 03690 Alicante, Spain

[2] Department of Computer Architecture, Faculty of Electronics, Telecommunications and Informatics, Gdansk University of Technology, Gabriela Narutowicza 11/12, 80-233, 03690 Gdańsk, Poland

Published online: 07 June 2023

algorithms), thus offering mobility with computing power. Nevertheless, the new features of mobile applications require greater performance, and the requirements are steadily increasing. Consequently, devices are becoming obsolete in a short time. In this way, the utility of the devices can be increased by offloading computation-intensive applications to a high-performance computing infrastructure hosted outside the mobile devices, normally to the cloud. The paradigm Mobile Cloud Computing (MCC) provides many more possibilities for computing complex applications in standard IoT devices such as mobile phones, wearables, or other embedded systems [2, 5]. Offloading the application workload running in the 'things' could save power consumption and bring greater computing resources. Recent technological developments are evolving towards this 'outsourcing' concept by deploying computing platforms in different layers of the network, not only at remote servers over the Internet. Moreover, there are a growing number of 'things' with processing capabilities in the surrounding environment of a mobile device or embedded system that can also be used to share the processing [6, 7].

Mobile and embedded systems are progressively incorporating connectivity and providing ready-to-use processing capabilities. Other computing resources are being installed at different network levels that work as key market drivers to improve the development of high-tech applications for advanced environments such as smart cities, autonomous vehicles, and advanced industrial systems. In this way, nearby clouds can be deployed in corporate facilities [8, 9] and/or on the mobile network [10, 11].

The fast development of wireless network technology has led to the emergence of fifth-generation (5G) networks, which offer faster data transfer speeds, lower latency, and higher capacity [12, 13]. These networks are becoming increasingly available due to the growth of IoT applications that require real-time processing of large amounts of data. One of the key advantages of 5G networks is their ability to handle a larger number of connected devices, which is critical for the development of IoT ecosystems. To further optimize the use of these networks, MCC has emerged as a promising paradigm that enables a more efficient and effective utilization of IoT devices [14]. By offloading computation tasks to the cloud, MCC paradigm can improve the performance of IoT devices and enhance the overall quality of service (QoS) provided by 5G networks. This paradigm is a promising approach to address critical challenges encountered by IoTs systems. These challenges are primarily related to privacy and security concerns, as well as to optimizing energy consumption. The adoption of this paradigm allows for efficient management of resources and data, enabling IoT systems to achieve higher levels of functionality and sustainability [15, 16].

However, to our knowledge, there is no comprehensive study or review of the MCC paradigm working in a full scenario with several available computing platforms along the network layers in a combined way. Consequently, the motivation of this work is to review how the MCC paradigm could be generalized to the current available computing platforms and to research the advantages and possibilities of architectures with multiple offloading options, as opposed to the traditional MCC scheme. Therefore, the main objective of this work is to investigate the advantages and possibilities of architectures with multiple offloading options as opposed to traditional MCC schemes and to understand how they can provide greater performance to compute advanced IoT applications.

The key contributions of this work can be summarized as follows:

- A study of the available architectures and computing platforms to outsource processing load and a review of this process using a combination of the computing layers of the available infrastructure.
- A demonstration of the benefits of offloading using all computing layers of the network by designing a proof-of-concept prototype that functions as an illustrative case study in which numerous tasks are involved with different characteristics and several scenarios.

The novelty of this proposal is the focus on generalizing the MCC paradigm to the available network computing platforms to increase the performance in computing advanced IoT applications. Other important aspects, such as privacy issues, are outside of the core of this research. However, these topics can be improved under the proposed framework because it is easier to manage security at nearby sites rather than remote clouds.

The paper is structured as follows: Sect. 2 provides an overview of the MCC paradigm, where the main operational concerns and platforms are described; next, in Sect. 3, the idea to extend the MCC concept is introduced. An example is used to explain how the network-assisted processing takes place; Sect. 4 discusses the benefits it provides; and finally, Sect. 5 draws the main conclusions and future research lines.

2 Mobile Cloud Computing overview

This section describes the potentials and limitations of the MCC paradigm, and after that, the possibilities for outsourcing the application workload are discussed, as well as the most relevant existing architectural design possibilities

for this area. Finally, some conclusions are drawn that justify the main contributions of this research.

2.1 MCC operational concerns

The MCC paradigm of computation extends the capabilities of devices in the execution of applications. In this way, it can be defined as an 'augmented device' system. The most common uses of this paradigm are mainly focused on extending the battery life of mobile platforms [17–21] without considering the versatility that remote computers can provide to extend the computing power of devices. However, recent work considers the increase in performance to be one of the most important contributions of MCC to mobile computing [22, 23].

This approach brings great potential to the development of the IoT paradigm, since it can provide computing power when and where necessary for connected things and turn them into smart things. In this way, it can be considered as a disruptive paradigm enabler of advanced applications. The two main disruptive changes can be summarized as (a) homogenization of device computing capabilities because they can execute applications regardless of their native hardware and (b) overcoming the limitations of mobile devices in the execution of advanced applications.

The concept of offloading computation is used to address the inherent problems of mobile computing in using other computing resources than the device itself to host the execution of mobile applications [22]. Of these challenges, the following are highlighted [2, 24–27]:

(1) *Using heterogeneous cloud resources* managing cloud resource utilization for many components and devices around the world is not a trivial undertaking. The tasks involved, such as partitioning resource-intensive components, virtual machine (VM) creation and migration, and monitoring the overall outsourced processes, can produce overhead for the cloud management system.

(2) *Elasticity of infrastructure providers* there may also be situations where there are more demands from MCC clients than available resources. This can cause cloud resource unavailability, service interruption, and performance degradation.

(3) *Unpredictable and long latency* response times and delays in server responses are difficult to predict. In offloading tasks from MCC devices, situations can occur where server response delays can increase. When network latency increases, the quality of user experience is decreased. However, better network protocols and new telecommunication technologies (such as, for example, promising 5G) could reduce this drawback.

(4) *Security and Privacy* this is a challenging issue in MCC because different devices use heterogeneous computing resources through a communication network. This is the main reason why private clouds (or other private deployed resources) are still a good choice for organizations, since they are deployed and managed inside them.

2.2 MCC platforms: solving challenges and related work

At the moment, new platforms with computing capabilities have been deployed at several layers of the network with different objectives: increasing security, keeping privacy, providing specialized resources, etc. [28]. The network has been labeled at two ends, called the 'edge' and 'core' ends [29–32]. The edge end is close to the data sources and users; the core end consists of the cloud servers. In general, edge computing aims to bring cloud resources and services to the edge of the network [31]. This concept aims to provide resources and services to the end user with minimal delay. Figure 1 shows a scheme of these different current offloading options along the communications network between the devices and the cloud servers.

The traditional conception of the MCC paradigm considers the cloud as the target platform for offloaded computation [2, 5, 22]. The Cloud is referred to as the computing infrastructure hosted on remote servers that can be accessed through the Internet. Challenges (1) and (2) are common in the management of cloud server resources [33]. With the MCC paradigm, this problem increases significantly because the volume of 'things' deployed is growing rapidly due to the expansion of the IoT [34]. Internet access through wide area networks (WANs) is the main culprit [Challenge (3)]. The network can behave unpredictably at any time due to different aspects. For this reason, it is difficult to provide a smooth scheduling of response times. Regarding security issues [Challenge (4)], offloading work to external servers is a critical task. Even in the case of private infrastructures, access to a WAN can be a source of security issues. Recent proposals have attempted to overcome the previous drawbacks by approaching cloud computing resources to mobile devices that will consume them [4, 35, 36].

Regarding the mobile edge computing (MEC) paradigm, the main efforts are directed towards reducing network latency by moving the computation and storage capacity from the core WAN to the edge network [37]. This goal can be achieved due to the evolution of communication base stations towards platforms capable of providing secure computing and IT services [38]. Thus, these resources can be used by connected mobile devices near stations to

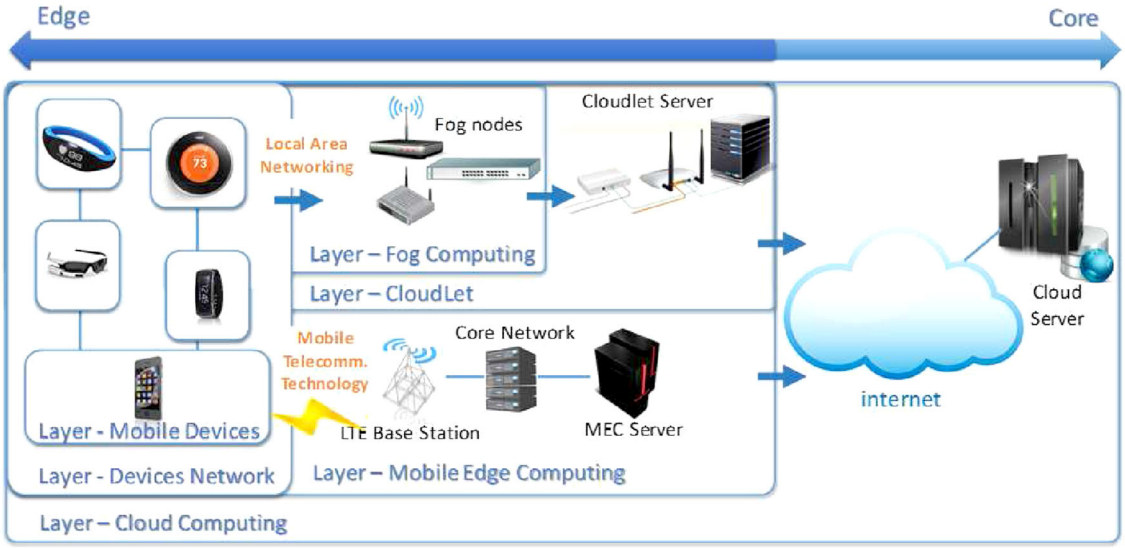


Fig. 1 Offloading platforms along the communication network

offload application tasks and execute part of the computing load. In turn, edge platforms can shift computation to traditional cloud servers to compute intensive calculations or centralized database access tasks [35].

MEC reduces Challenge (2) because it can offer additional resources for outsourcing the computation load of advanced applications and then increase the elasticity of cloud computing. Furthermore, MEC platforms can be deployed in numerous locations and build a dense network where necessary, providing flexibility [37]. This offloading process is transparent to the user. Mobile devices maintain services by sharing the processing between the MEC and the cloud. Challenge (3) is also addressed. In this case, the response times of the offloaded tasks are reduced due to the higher bandwidth and proximity of the MEC platforms.

On the contrary, Challenges (1) and (4) remain. Decentralization of infrastructure adds heterogeneity to cloud resources and increases management costs. The security issue remains unsolved. In this case, a distributed security strategy is required to consider the entire system and the different edge platforms. Furthermore, a standardized environment is required to adequately address this problem and specify how the different elements of the architecture can collaborate with each other [39].

The cloudlet infrastructure is a step forward towards bringing cloud resources closer to mobile devices. The physical proximity of the cloudlet simplifies the challenge of meeting the required bandwidth and provides improved performance in response times [40–42]. However, the limited resources of the cloudlet infrastructure negatively affect performance with an increasing number of user devices. Hence, the cloudlet must move this additional processing to the core cloud systems to meet the requirements. As mentioned, cloudlet infrastructure is usually deployed within a local area network (LAN) and can be accessed wirelessly [7]. The bandwidth of the wireless LAN is typically two orders of magnitude higher than the bandwidth of the wireless Internet available to a mobile device [43].

The deployment of cloudlet is designed specifically to address Challenges (2) and (3) and to provide scalable resources [44]. Therefore, it can be used to improve the QoS of interactive applications in citizen-centric environments such as smart cities [45]. Cloudlets are decentralized, widely distributed, and self-managing. Hence, they are not as efficient as core cloud computing, because the resources are sparse in a wide area (for example, each city district could have its own cloudlet). Therefore, Challenge (1) is not well addressed by cloudlets. Concerning the security issue [Challenge (4)], the security and privacy vulnerabilities inherent to the communications across the network remain. However, the cloudlet nodes are closer to the users, and the platforms are owned by the companies providing the service. In this sense, it is less risky to compute the data in the cloudlet compared to a remote server.

The next step is fog computing, where the closest network devices can provide a computing aid to the applications. This infrastructure provides real-time capabilities to devices that minimize security risks and offloading distance and communication latency [8, 46]. It is inefficient to transmit all the data acquired by the sensors to the cloud for processing and analysis. Thus, this paradigm has been designed specifically to provide computing capabilities to nearby IoT infrastructures, such as sensors or embedded devices. It has been proven that the closer the cloud

computing resources are to the devices, the more useful they are and the better the performance achieved. However, this novel paradigm has a lack of standards related to several aspects such as communication, data storage models, and interoperability of service discovery.

These mini-clouds deployed by fog nodes have a high impact on Challenges (2) and (3) issues because they provide the first line of computing [47]. Many of the application tasks can be solved at this level, and the others can be offloaded in turn to remote clouds. In this manner, unpredictability in response times does not disappear; however, it can be reduced significantly. The management of heterogeneous cloud resources [Challenge (1)] remains. Fog nodes can be extremely heterogeneous with no possibility of being scaled. Wireless network security [Challenge (4)] is a significant problem in fog networking because wireless communication is predominant in fog networking, even in edge computing, in general. Additionally, fog nodes are in the environment of the end user and can collect more sensitive information than a remote cloud. This aspect adds greater concern regarding the privacy issue [48, 49].

The final step in providing infrastructure for offloading applications consists of the sharing of computation tasks among neighboring connected devices. These devices become a collaborative environment called an 'ad hoc' cloud [50, 51]. This peer-to-peer computing allows applications to run cooperatively on devices that are in close proximity. Therefore, the end devices can be considered as edge computing platforms. For example, a mobile device can process the data from a near sensor [29]. This computational paradigm is highly scalable, depending on the available devices. However, the management of the computing platforms can be difficult and thus Challenge (1) remains unsolved or can even be increased. It is necessary to define efficient discovery services, user agreements, and other aspects such as task scheduling or resource pricing. However, this configuration provides considerably more flexibility to address computation requirements. Data can be processed over the set of computing devices in its surroundings without the requirement of Internet access or available cloud resources. Hence, it can ease Challenge (2) by providing elasticity in access to cloud resources. Direct communication between devices also provides solutions to Challenge (3). The response time and latency are highly predictable because there are no network congestion problems. However, the availability of the computing power of existing devices can limit the resulting performance. Regarding security [Challenge (4)], this computational model better protects data because it is not transmitted to cloud servers or external platforms for processing. However, in a heterogeneous and variable network topology, it is difficult to support effective security and privacy mechanisms [52]. Other additional challenges arise, such as power consumption restrictions and access permissions to the devices.

Table 1 summarizes the main features of each platform level related to the offloading process and the purposes of this work. The data have been extracted from the references cited in this section.

2.3 Findings and novelty of the proposal

After reviewing the design possibilities and development of architectures for offloading mobile computation, some findings can be drawn that justify and summarize our contributions to previous MCC architecture designs:

- The trend to improve the overall performance of complex applications deployed in distributed environments (such as IoT and mobile applications) is to identify methods of using the networks and the Internet to provide additional computing power to mobile devices and 'things'. The proposals are intended to add computing resources at several layers of the network and to move increasing computing capabilities closer to data sources and users.
- Several approaches have been proposed to offload application tasks that take advantage of technological progress in communication, embedded systems, cloud computing, and application development.
- In the new paradigms and applications that deploy connected 'things' and embedded systems, mobile ad hoc clouds can be an effective option to share processing tasks. These mechanisms can be considered to improve the possibilities of outsourcing work.

This work proposes an extended architecture for the MCC paradigm to enhance the offloading process based on network resources. This approach combines ad hoc clouds, fog nodes, cloudlets, MEC platforms, and cloud servers to improve the flexibility of computations and determine the best way to execute applications.

3 Extended MCC architecture

3.1 General framework

In this section, the architecture for performing network-assisted processing is introduced. This architecture aims to reduce the impact of previous challenges for high-demanding applications that can be executed along the network. In order to verify its functioning, a prototype and study case of the multi-layer architecture is described in this section. The prototype is validated by running a complex application in a realistic operating environment.

Table 1 Offloading features of each network platform

Feature/Platform level	Ad hoc cloud	Fog computing	Cloudlet	MEC	Cloud
Computing power	Low	Low	Medium Variable	Medium Variable	Very high
General availability[a]	Low	Low	Medium	High	High
Offline availability[b]	Yes	Yes	Yes	No	No
Latency	Very Low	Very Low	Low	Medium	High
Bandwidth	High	High	High	Medium	Low
Volume of users at a time	Very Low (a few)	Low (<10)	Medium (<100)	High (<1000)	Very high (≥ 1000)
Scalability[c]	Very low	Low	Low	Medium	High
Deployment environment	Decentralized Devices/things	Decentralized Network devices	Decentralized Local server machine	Decentralized Base stations	Centralized Large data centers
Fault tolerance	Very Low	Low	Medium	Very High	Very high
Management	User-management	Supervised	Self-management/ Supervised	Professionally, 24×7 operator needed	Professionally, 24×7 operator needed

[a]The general availability feature is understood as there is an enabled platform of this kind

[b]Offline availability refers to the offloading capability when there is no access to the Internet or a communication network

[c]This feature describes the ability to increase computing resources if required

The purpose of the experiments performed is to illustrate the flexibility provided by the different offloading options and their implications for addressing the challenges. The results demonstrate that only the availability of several offload layers can overcome the operating conditions imposed by the application scenarios.

The application used consists of computing the 'Autonomous Vehicle Driving' (AVD-App) executed on board a car. The operating environment is an urban area of a Smart City where several autonomous vehicles are circulating at the same time on the streets. This will be common in the near future, where vehicles will be required to make decisions in real time and provide information to the city to facilitate traffic management in a similar time frame. This application requires a huge amount of computational power because there are numerous driving aspects involved and obstacle trajectories must be calculated and verified at every moment. To address this problem, one design option could be to provide the embedded car system with sufficient computing resources to complete all computations. However, in this scenario, new application updates and capabilities could exceed the system's resources in the future. Another option is to offload part of the processing cost to the cloud. This option requires Internet connectivity and a powerful cloud system to provide service to the entire vehicle fleet. Between these extremes, the proposed approach consists of combining the computing resources of the network architecture to perform all computations. This approach provides flexibility and robustness by increasing the computation options at several computing layers. In this work, a simplified version of the AVD-App is described; therefore, the non-relevant details for this work are not mentioned.

The operation of the framework for network-assisted processing is defined by three key factors: (i) application, (ii) infrastructure, and (iii) criteria.

3.1.1 Specification of the application

Firtly, the applications suitable to be processed by this architecture must be ready for the proposed workload. This means that the application workload can be partitioned and distributed along the network to be processed by different layers.

In this way, let Γ_{Appl} be the set of tasks of the application and $\{t_1, t_2, \ldots, t_n\}$ the application tasks to be computed, as shown in Eq. 1:

$$\Gamma_{Appl} = \{t_1, t_2, \ldots, t_n\}. \quad (1)$$

The AVD-App can be decomposed into common tasks according to the findings of recent research works [53–56] in a possible Smart City context. The simplified set of tasks for the AVD-App used in this study is shown in Fig. 2. The set of tasks is defined by the set $\Gamma_{Appl} = \{t_1, t_2, \ldots, t_7\}$. The application tasks are running continuously while the vehicle is operating.

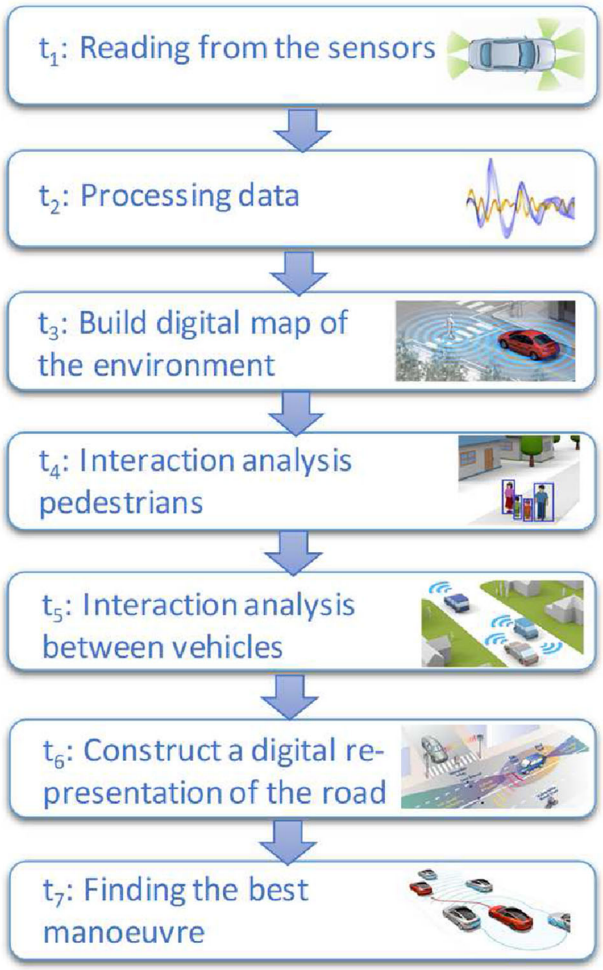


Fig. 2 Autonomous vehicle driving application decomposition

A brief description of the tasks is as follows:

t_1 *Readings from sensors* this task manages the raw data acquired by the sensors and prepares them for processing. Sensors can be autonomous devices or part of embedded systems. In some cases, preprocessing (digitalizing, filtering, enhancing, and sampling) might be required to obtain the correct data. The volume of information generated by this stage depends on the number of sensors and their nature. Data are derived from several sensors including video cameras, LIDAR (Laser Imaging Detection and Ranging) scanners, radar, and GPS (Global Positioning System). The range of vehicle sensors determines the maximum speed of the vehicle. Thus, the greater the sensor range, the better the knowledge of the environment and the greater the speed the car can achieve. Sensors are considered to have an effective perception of the environment of ten meters around the vehicle.

t_2 *Processing data* from the results of the previous task, in this stage, the data are processed to recognize objects and elements of the environment (pedestrians, traffic signals, and other vehicles).

t_3 *Build digital map* in this stage, the objects identified in the previous task are combined to build a digital map of the environment. That is, locate pedestrians, obstacles, and other vehicles. Additionally, a road map is created to identify the limits of the road. In this way, as the vehicle travels on the road, a digital map is created to transform the continuum of the environment into a digital representation of the environment, which is the essential space for planning.

t_4 *Interaction with pedestrians* this task aims to determine the interaction between the autonomous vehicle and pedestrians. It predicts the intention of the participants in human traffic, calculated separately for each pedestrian. It is assumed that intentions do not change over time.

t_5 *Interaction with vehicles* this task analyzes the interactions between vehicles and aims to predict the movement of other vehicles. It is also calculated separately for each vehicle detected.

t_6 *Digital representation of the road* from the previous information, in this task, the driving corridors are constructed according to the predicted motion of the dynamic obstacles (pedestrians and other vehicles) and the presence of static objects.

t_7 *Determine best manoeuvre* determining the best maneuver to perform the movement is based on calculating the relative positions of the other participants in the environment at the time of making a decision, estimating the risk of possible situations, and making a decision about the best geometric path for the vehicle to follow.

Each previous task has a specific computational cost which can vary depending on the complexity of the environment and the participants involved (obstacles, pedestrians, and other vehicles). Tasks 3 and 6 are intensive in multimedia processing; Tasks 2, 4, and 5 are intensive in signal processing. Task 7 is intensive in both types of specific processing. To reduce computational cost, several simplifications and assumptions can be made. For example, the system can consider a simple representation of vehicles as rectangles and obstacles as circles. However, in this case, close proximity motions cannot be performed due to the lack of accuracy in the approximation [54]. The proposed architecture can avoid these simplifications to improve reliability and allow the system to operate in more complex situations.

The cost of computing the entire application is also an important determinant of the speed of the vehicle. In this

case, the sensor range and frequency of decision determine the maximum speed at which the vehicle can move. In this prototype example, it is considered valid to make a maneuver decision every ten meters traveled (except for sudden movements).

3.1.2 Infrastructure

Several computing layers and platforms can be used to process the application tasks. In Eq. 2 let L represent the collection of computing layers that can be accessed through the network design, and let L_0 and L_{cc} represent the network's ends, where L_0 represents the edge device (the closest one to the IoT device) and L_{cc} represents the last available architecture layer, with a remote cloud-computing data server:

$$L = \{L_0, \ldots, L_{cc}\}. \tag{2}$$

In general, the available computing layers are variable according to numerous aspects, such as the execution environment, the available infrastructure, and the configuration options. In rich contexts, there may be several computing platforms to process the workload. In other cases, the offloading options are limited. In between, there are several computing layers available to perform the processing at different levels. For example, in an autonomous vehicle application context, L_1 can be the network composed of several nearby vehicles, L_2 can be the layer formed by the city traffic infrastructure or cloudlets deployed, and L_3 can be the mobile edge computing infrastructure behind the communication network. Other infrastructure configurations can be set up for this application.

Each layer (L_j) has a set of computing platforms which can be heterogeneous with different processing capabilities and abilities according to their characteristics. Thus, layer L_j has m_j computing platforms (see Eq. 3), where $j \in \{0, \ldots, cc\}$ and $m_j > 0$. The set $\{0, \ldots, cc\}$ represents the available computing platforms that span from mobile devices to cloud servers, thus providing a comprehensive range of resources for distributed computation.

$$L_j = \{p_{j1}, p_{j2}, \ldots, p_{jk}\}, \tag{3}$$

where p_{jk} is the platform $k \in \{1, \ldots, m_j\}$ of the layer j.

The different layers of the network can be deployed in sequence or in a parallel configuration, where each computing platform of a layer can execute the services of the upper layers and provide services to several elements of the lower layers.

The infrastructure of the case study is as follows (Fig. 3).

In the described environment, there can be five computing layers for computing the application:

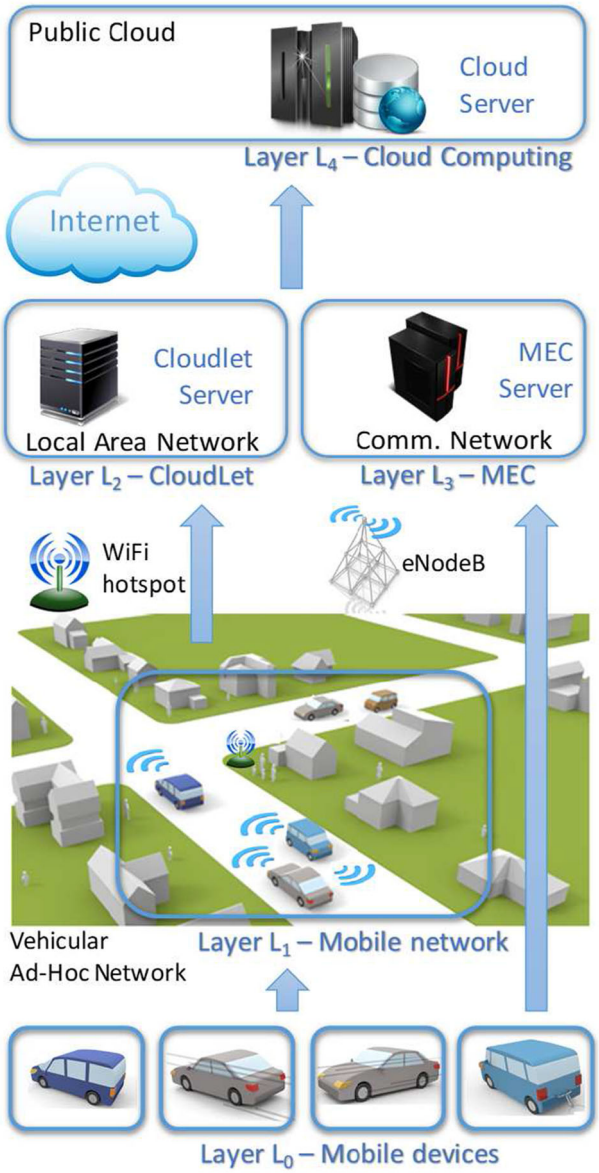


Fig. 3 Prototype of the network architecture for autonomous vehicle driving application

L_0 *vehicle* it is the computing system built into the vehicle. This layer consists of each isolated vehicle.

L_1 *set of nearby vehicles* this layer consists of the VANETs existing at a given time. A VANET provides wireless communication between a set of moving vehicles [56]. The VANET vehicles can exchange information and share computing resources to offload processing tasks to each other. This layer is an ad hoc cloud.

L_2 *district cloudlet* in the planned configuration, each city district has a cloudlet. Therefore, this layer has the set of cloudlets of the city. Each cloudlet establishes a LAN where vehicle systems and VANETs can interact. That is, the cloudlet is a private cloud that provides additional processing capabilities to the vehicles. Furthermore, in the described application context, these resources can be used for managing other Smart City infrastructures such as smart traffic signals, parking meters, or urban furniture.

L_3 *mobile edge computing* this layer has the computing resources available from the communications network. This layer can be optional and accessible through mobile communication technologies such as LTE. If it exists, it can provide additional computing resources to the systems. The urban area covered by each base station is approximately 1.5 km in radius. The deployment of dense networks can provide more base stations to provide processing power to the city [57].

L_4 *cloud computing* this last layer provides the cloud computing resources of the architecture. It can be a public cloud hosted outside the city and can be accessed through Internet protocols.

The available computing layers depend on the communication facilities of the vehicles. For local area networking, the available layers will be $L = \{L_0, L_1, L_2, L_4\}$ and for LTE connections: $L = \{L_0, L_3, L_4\}$. Additionally, there are devices that support both types of communication. In these cases, all layers will be available to them.

The size of each level is variable because it is based on traffic conditions. The number of computing platforms depends on the zone, the existing vehicles, and the circumstances of the city. In addition, computer platforms can be heterogeneous at each level. Thus, vehicles can have different features, cloudlets can have processor capabilities according to their average workload (i.e., the most powerful for city center and highest population districts), and the MEC and cloud computing infrastructures can be adapted to the computing requirements of the city (i.e., most powerful for rush hours and less at night or on weekends). All of these are examples of the computing flexibility of the architecture. In this example, a generic approach with all available layers has been described. Other configurations are possible with a reduced number of layers. There can be different cases where driving conditions evolve; for example, zones without traffic (no layer 1), districts without cloudlet (no layer 2), and areas outside of the communication coverage (no layers 3 and 4).

The list of described platforms represents the specific network architecture of the device and indicates the possible offloading of work at a given time. This list can vary over time, depending on the context of the application. For example, if the mobile device is moving, the available infrastructure can change.

3.1.3 Criteria

The third key factor is related to why application tasks are offloaded to different computing platforms. The criteria determine what the determinant execution costs are involved in the computation of the application.

The execution cost can be any of the different performance aspects that are involved in the execution of a task on a platform. Thus, let Ω be the set of performance aspects to consider (see Eq. 4). This covers a range of variables such as computing cost, power consumption, and monetary expense, among others.

$$\Omega = \{\text{computing cost, power consumption, money,}\ldots\}. \tag{4}$$

The selection of criteria is based on multiple configurations, depending on the type of application, execution restrictions, or operating conditions. A single aspect or a combination of several aspects of Ω can be considered depending on the specific requirements of the application.

The underlying idea is that each of the available platforms could have a value for each of them. This information is used to feed the scheduling algorithm along the network and to decide where tasks are computed. In this way, the application could search for the best option to meet the requirements.

Regarding the management of the offloading process and the scheduling method, there are many recent contributions to outsourcing the workload in multi-tier network architectures [58, 59]. There is a multitude of works that propose a method for distributing workload, for example, our recent research on IoT applications [60]. A middleware layer might be necessary to allow communication of heterogeneous computing platforms, to include manager modules, as well as other interesting services such as filtering, discovery, and availability services for receiving and executing tasks [61]. This layer can be installed within the devices or deployed around some edge computing platform.

To avoid scheduling overhead in complex environments with multiple offload options, a prediction method based on performance estimations can be used. These data are based on historical performance measurements and are continuously updated. However, this solution cannot be extended to all applications and situations; it is valid when the possibilities can be clearly defined, as indicated in this case. In addition, there may be platforms with specific capabilities that provide services to many applications and

Table 2 Computing cost (units in ms)

Task	Device	Mobile network	Cloudlet	Base station	Cloud server
T_1	50	12.5	5	5	2.5
T_2	250	62.5	25	0.125	12.5
T_3	200	50	0.1	20	10
T_4	100	25	10	0.05	5
T_5	100	25	10	0.05	5
T_6	250	62.5	0.125	25	12.5
T_7	50	12.5	2.5125	2.5125	2.5

Table 3 Generated data by each task in kB/s

Task	Generated data
T_1	1024
T_2	256
T_3	512
T_4	128
T_5	128
T_6	1024
T_7	5

allow acceleration of the processing of specific types of tasks. For example, GPUs can be installed on cloudlet servers to accelerate multimedia algorithms. In this manner, the granularity of this calculation is the cost of computing each task.

This work is focused on the multi-tier network architecture and exploits previous research for scheduling of the tasks. In this way, it is necessary to define the performance of the available layers of the architecture. The main aspect considered is the time delay (including computing time and communication costs). That is, $\Omega = \{\text{computing cost}\}$. The same principle of flexibility that was followed throughout this work can be similarly applied to other performance metrics.

3.2 Simulation and discussion

In this section, a simulation of the proposed architecture is performed according to the previous application environment. The main objective of this section is to demonstrate the flexibility of the architecture and the different possibilities to outsource the workload along the network layers and to overcome difficulties as they arise. The simulation is carried out to obtain different performance results according to the environmental conditions and the working scenarios. It is designed as a five-layer architecture, as illustrated in Fig. 3 considering built-in devices, mobile network, cloudlet infrastructure, communication base station servers, and remote cloud computing. Other layers can be incorporated under the same principles described, providing extra flexibility and computing possibilities to the architecture.

3.2.1 Simulation setup

Many previous works present experiments and simulations of offloading applications between several layers of the architecture [6, 44, 51, 59, 61]. In these investigations, performance data is provided that demonstrate the benefits of the outsourcing process. The results of these works have been considered to estimate the cost of application and network in terms of time delay. To introduce heterogeneity to the computing layers, the cloudlet platform is equipped with GPU devices, which accelerates the multimedia tasks, and the base station platforms have been equipped with DSP devices, which accelerate the signal processing tasks.

Based on the experiments and simulation of the previous works, the following tables show the values of the performance of the framework. The calculation of the computing cost of each platform is given in Table 2. The data generated by each task are shown in Table 3.

This application is executed continuously while the autonomous car is operating. When a driving maneuver is performed, new data acquired by Task 1 becomes available. Thus, the tasks produce a continuous data stream per second. For example, the data flow acquired by the sensors in Task 1 is 1 MB/s.

Finally, the communication costs are listed in Table 4. The cost of these communications usually depends on the volume of data transmitted. *Short Range Communication* for vehicular networking, WiFi communication with cloudlet platforms and LTE with base stations are chosen. The devices and cloudlets are in a LAN. The base stations and cloud servers are on a WAN. The costs are extracted from the experiments on communication performance analysis carried out by Kaya et al. [62] and da Mata and Guardieiro [63], as shown in Table 4. Symmetric communication is considered in all cases.

The costs of computation and communication can change with time. To simulate different situations, several scenarios have been defined: (A) normal scenario with fluid flow traffic, (B) busy street, (C) traffic congestion in the city district, (D) rush hour, and (E) congested communication network.

The normal scenario (A) corresponds to the working environment described by Tables 2, 3 and 4. The busy street (B) is the situation produced in which many elements, such as other vehicles, traffic signals, and pedestrians, are located near the autonomous car. In this context, more items imply that the computing costs of the tasks are increased. The simulated data consider a cost double (×2) the data depicted in Table 2 for the computing cost of this scenario, except for the cloud server (×1.4), due to its higher response capacity to absorb more processing. Traffic congestion in (C) for the city district corresponds to the same scenario as in (B); however, in this case, this situation affects a city district. Thus, the cloudlet deployed in this district has an additional delay in processing the tasks. An increase of (×10) is estimated for the computing cost of the cloudlet represented by Table 2 to simulate this scenario. The cost of the other platforms remains at (×2). The rush hour scenario (D) affects a larger area of the city. In this case, the communication cell of the base station has many users using services. Both the cloudlet and base station server produce an additional delay in computing the tasks. An increase of (×10) the computing cost of the cloudlet and base station server is estimated. Finally, the scenario of congestion of the communication network (E) implies a higher communication cost for WANs, that is, base stations and cloud servers. In this context, the communication cost with these platforms is trebled (×3). Table 5 displays the different scenarios of this simulation and their effects on the cost functions with respect to the cost shown in Tables 2 and 4.

The architecture scheduling algorithm that determines where and when to offload the computation considers the costs and scenarios depicted in Tables 2, 3, 4, and 5 to build a look-up table to predict the behavior of the network platforms and coding the offloading decision in each case. Table 6 summarizes the simulation parameters.

3.2.2 Simulation results

From the specifications described in the previous sections, numerical results were obtained after simulating the autonomous vehicle operation in the different scenarios proposed. The experiments carried out allowed us to identify the best options to execute the application and the possible decisions to offload the work in each case. Minor random changes can be included to the cost data to validate the architecture in a more realistic functioning. These results can improve the knowledge of the scheduling offload method.

Figure 4 shows possible configurations for offloading tasks among network platforms. This figure gives a graphic representation of the sequence of platforms where the application is executed. Other offloading combinations can be considered on the basis of the performance aspect analyzed and the scheduling function.

In the next section, we will describe the behavior of the multi-layer architecture in the different designed scenarios. The results of the simulations are presented in Table 7.

Table 4 Communication cost

Communication cost	Device/mobile network	Cloudlet	Base station	Cloud server
Device	0.2143	0.4286	0.7812	2.2177
Cloudlet	0.4286	–	–	1.7814
Station	0.7812	–	–	1.7814
Cloud	2.2177	1.7814	1.7814	–

Table 5 Computing and communication cost in working scenarios

Scenario	Device cost Comp.	Comm.	Cloudlet cost Comp.	Comm.	Base station cost Comp.	Comm.	Cloud server cost Comp.	Comm.
(A) Normal	1×	1×	1×	1×	1×	1×	1×	1×
(B) Busy street	2×	1×	2×	1×	2x	1×	1.4×	1×
(C) Traffic congestion	2×	1×	10×	1×	2×	1×	1.4x	1×
(D) Rush hour	2×	1×	10×	1×	2×	1×	1.4×	1×
(E) Network congestion	1×	1×	1×	1×	1×	3×	1×	1.7×

Table 6 Simulation parameters

Application	Autonomous Vehicle Driving
Tasks	$\Gamma = \{$Readings from the sensors, Processing data, Build digital map, Interaction with pedestrians, Interaction with vehicles, Digital representation of the road, Find best manoeuvre$\}$
Network	Multi-layer architecture
Available layers of the architecture	$L = \{$devices, ad hoc cloud (VANET), cloudlet, base stations, cloud server$\}$
Performance aspects	$\Omega = \{$computing cost$\}$
Specialized hardware features	Cloudlet equipped with GPUs; base stations equipped with DSPs
Scheduling algorithm	Prediction based. Performance estimations from Tables 2, 3, 4, and 5

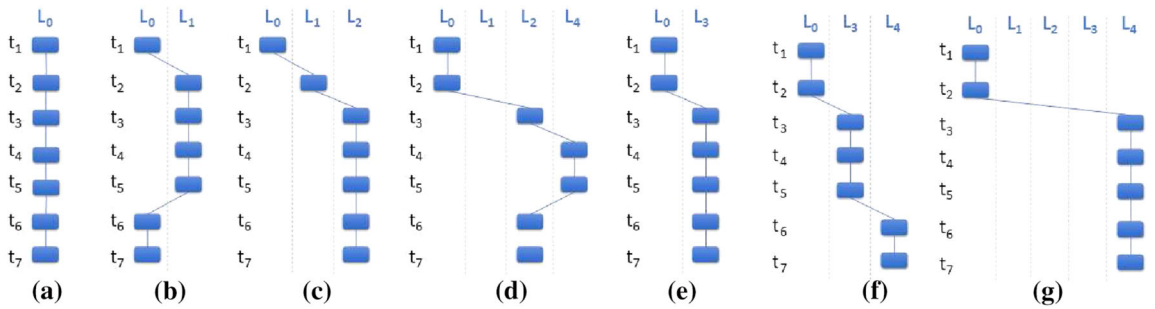


Fig. 4 Configurations for offloading the tasks among the network platforms. **a** Only device, **b** Device–mobile network, **c** Device–mobile network–cloudlet, **d** Device–cloudlet–cloud server, **e** Device–base station server, **f** Device–base station server–cloud server, and **g** Device–cloud server

In the normal scenario, the computing cost of processing the entire application in the built-in device of the car was 1000 ms. That is, the autonomous system made a movement decision per second. This cost sets a maximum speed of 10 m/s (36 km/h). This is a normal speed for city streets. The MCC paradigm for this application improved this time, allowing faster speeds using external platforms to outsource computing. In the previous table, the configurations with improved performance are highlighted. In these cases, the speed of the autonomous car could be increased due to the reduced time required to execute each iteration of the application.

4 Analysis and discussion

The results obtained in the previous section confirm two important findings for the MCC paradigm. First, the multilayer architecture allowed higher flexibility in offloading the computations along the available network layers to meet the requirements, not only with regard to the time delay but also other performance aspects such as monetary cost and power consumption. The results in Table 7 illustrate the minimum cost configurations for each scenario. For each of them, there are some configurations that can reach the desired performance. Of course, this is just an example, but shows that the network-assisted processing concept provides clear advantages over cloud-targeted outsourcing methods. Second, the proposed multilayer architecture provided additional opportunities to improve application performance and, thus, better regulated the capabilities of mobile devices and connected 'things'. In all defined scenarios, the achieved speed of 10 m/s was improved using the different offload options. The experiment also demonstrated a comparison with other proposals that use a specific computing platform for offloading. None of the known platforms alone was sufficient to provide an effective response to the problem in all scenarios. However, the communication cost of the network usage must also be considered in the overall performance. In many cases, performance decreased due to the delay in the outsourcing process along the network platforms.

In general, this approach contributes to improving the overall operation of the MCC paradigm. Table 8 describes the main points to address the operational concerns of the MCC described previously.

Table 7 Simulation results for the offloading configurations in the working scenarios

Scenario	Performance: time delay (ms) Computing	Communication	Total	Maximum speed
(A) Normal scenario				
(a)	1000.0	0.0	**1000.0**	10 m/s (36 km/h)
(b)	512.5	246.9	**759.4**	13.1 m/s (47.4 km/h)
(c)	135.2	276.4	**411.7**	24.2 m/s (87.4 km/h)
(d)	312.7	1480.0	1792.7	5.5 m/s (20.0 km/h)
(e)	347.6	203.9	**551.5**	18.1 m/s (65.2 km/h)
(f)	335.1	439.0	**774.1**	12.9 m/s (46.5 km/h)
(g)	335.0	578.8	**913.8**	10.9 m/s (39.3 km/h)
(B) Busy street scenario				
(a)	2000.0	0.0	2000.0	5 m/s (18 km/h)
(b)	1025	246.9	1271.8	7.8 m/s (28.3 km/h)
(c)	270.5	276.4	**546.9**	18.2 m/s (65.8 km/h)
(d)	619.5	1480.0	2099.5	4.7 m/s (17.1 km/h)
(e)	695.2	203.9	**918.9**	10.8 m/s (39.1 km/h)
(f)	661.2	439.0	1100.2	9.0 m/s (32.7 km/h)
(g)	649.0	578.8	1225.8	8.1 m/s (29.3 km/h)
(C) Traffic congestion scenario				
(a)	2000.0	0.0	2000.0	5 m/s (18 km/h)
(b)	1025	246.9	1271.8	7.8 m/s (28.3 km/h)
(c)	679.7	276.4	**956.1**	10.4 m/s 37.6 km/h)
(d)	668.7	1480.0	2148.7	4.6 m/s (16.7 km/h)
(e)	695.2	203.9	**918.9**	10.8 m/s (39.1 km/h)
(f)	661.2	439.0	1100.2	9.0 m/s (32.7 km/h)
(g)	649.0	578.8	1225.8	8.1 m/s (29.3 km/h)
(D) Rush hour				
(a)	2000.0	0.0	2000.0	5 m/s (18 km/h)
(b)	1025	246.9	1271.8	7.8 m/s (28.3 km/h)
(c)	679.7	276.4	**956.1**	10.4 m/s 37.6 km/h)
(d)	668.7	1480.0	2148.7	4.6 m/s (16.7 km/h)
(e)	1552.2	203.9	1775.9	5.6 m/s (20.2 km/h)
(f)	1023.0	439.0	1462.0	6.8 m/s (24.6 km/h)
(g)	649.0	578.8	1225.8	8.1 m/s (29.3 km/h)
(E) Communication network congestion				
(a)	1000.0	0.0	**1000.0**	10 m/s (36 km/h)
(b)	512.5	246.9	**759.4**	13.1 m/s (47.4 km/h)
(c)	135.2	276.4	**411.6**	24.2 m/s (87.4 km/h)
(d)	312.7	1480.0	1792.7	5.5 m/s (20.0 km/h)
(e)	347.6	611.7	**959.3**	10.4 m/s (37.5 km/h)
(f)	335.1	1302.8	1637.9	6.1 m/s (21.9 km/h)
(g)	335.0	976.8	1311.8	7.6 m/s (27.4 km/h)

Bold indicates the cases in which the latency is $\leq$ 1 second

5 Conclusions

In this work, a distributed architecture for outsourcing computation tasks is described. This approach uses all available platforms along the network. In this manner, the proposed model configures an extended MCC paradigm, which allowed the outsourcing of the workload to other platforms. These platforms were separately analyzed to improve processing in previous work, and the main challenges were identified. In this research, it was proposed to use the entire available infrastructure to enhance the MCC paradigm beyond mobile computing applications and to leverage the deployed resources along the network: ad hoc clouds, fog nodes, cloudlets, MEC platforms and cloud servers.

Table 8 Extended MCC architecture contributions

Challenge	Extended MCC architecture
(1) Leveraging heterogeneous cloud resources	To the extent that cloud resources are becoming less necessary to compute IoT applications, the cloud management overhead is reduced
(2) Elasticity of infrastructure providers	The generalization of the MCC paradigm to the available infrastructure along the network provides higher flexibility to the offloading process. It allows the architecture to find the best option for improving the performance and meeting the application requirements
(3) Unpredictable and long latency	The availability of several offloading layers contributes to better define the response times. The response delay can be reduced by taking advantage of nearer computing platforms
(4) Security and Privacy	In principle, using computing platforms close to where data are generated does not expose data to potential security threats when communicated through the network. Additionally, some of these platforms can be within the same organization for private uses, avoiding risks of third party platforms

The main advantage of this proposal is the increase in flexibility in processing compute-intensive applications, which makes the system more resilient to environmental changes. The aim of the experiments carried out is to only make visible this flexibility provided by the different offloading options (Fig. 4) and their implications on the overall computing cost (Table 7). The performance data were acquired from previous research works on MCC and offloading frameworks, and from our own estimations. Other source data could produce different performance results. However, the key aspect of the simulations performed was not the specific cost obtained; instead, it was the flexibility and offload possibilities that the proposed multilayer architecture provided. The results show that only the availability of several offload layers can overcome the operating conditions imposed by the application scenarios. This experiment also shows a comparison against other proposals that use a specific computing platform for offloading, since none of the known platforms, by itself, could be sufficient to provide an effective response to the problem in all cases.

The approach presented in this work addresses the problem of providing additional computing power to mobile and embedded devices and contributes to some extent to address the operational concerns of the MCC (Table 8). The major incidence of this architecture occurs in resolving Challenges (2) and (3) due to the higher flexibility provided.

This research provides a wide variety of research lines for future work. In our opinion, these research lines can be classified into two sets: first, the research on interoperability of platforms and discovery of computing services to facilitate the distribution of processing along the network, and second, the design of efficient scheduling methods for outsourcing the work to multiple available computing platforms, especially when the device is functioning in a dynamic environment. In this set are located the QoS management and the performance prediction of the network.

Author contributions All authors were involved in the foundation items. All authors wrote the paper and read and approved the final manuscript.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature. This work was supported by the Spanish Research Agency (AEI) (DOI 10.13039/501100011033) under Project HPC4Industry PID2020-120213RB-I00.

Data availability Enquiries about data availability should be directed to the authors.

Declarations

Conflict of interest The authors declare that they have no competing interests.

References

1. Mell, P., Grance, T.: The NIST Definition of Cloud Computing. Technical Report NIST Special Publication (SP) 800-145. National Institute of Standards and Technology (2011). https://doi.org/10.6028/NIST.SP.800-145
2. Sanaei, Z., Abolfazli, S., Gani, A., Buyya, R.: Heterogeneity in mobile cloud computing: taxonomy and open challenges. IEEE Commun. Surv. Tutor. **16**(1), 369–392 (2014). https://doi.org/10.1109/SURV.2013.050113.00090

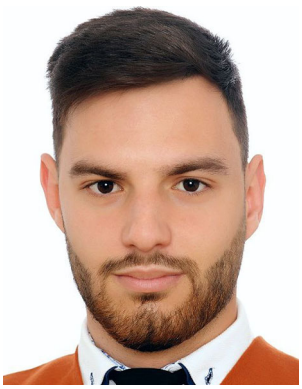
3. Cavalcante, E., Pereira, J., Alves, M.P., Maia, P., Moura, R., Batista, T., Delicato, F.C., Pires, P.F.: On the interplay of Internet of Things and Cloud Computing: a systematic mapping study. Comput. Commun. **89–90**, 17–33 (2016). https://doi.org/10.1016/j.comcom.2016.03.012
4. Hamdan, S., Ayyash, M., Almajali, S.: Edge-computing architectures for Internet of Things applications: a survey. Sensors **20**(22), 6441 (2020). https://doi.org/10.3390/s20226441
5. AlAhmad, A.S., Kahtan, H., Alzoubi, Y.I., Ali, O., Jaradat, A.: Mobile cloud computing models security issues: a systematic review. J. Netw. Comput. Appl. **190**, 103152 (2021). https://doi.org/10.1016/j.jnca.2021.103152
6. Colom, J.F., Mora, H., Gil, D., Signes-Pont, M.T.: Collaborative building of behavioural models based on Internet of Things. Comput. Electr. Eng. **58**, 385–396 (2017). https://doi.org/10.1016/j.compeleceng.2016.08.019
7. Yi, G., Kim, H.-W., Park, J.H., Jeong, Y.-S.: Job allocation mechanism for battery consumption minimization of cyber–physical-social big data processing based on mobile cloud computing. IEEE Access **6**, 21769–21777 (2018). https://doi.org/10.1109/ACCESS.2018.2803730
8. Shaukat, U., Ahmed, E., Anwar, Z., Xia, F.: Cloudlet deployment in local wireless networks: motivation, architectures, applications, and open challenges. J. Netw. Comput. Appl. **62**, 18–40 (2016). https://doi.org/10.1016/j.jnca.2015.11.009
9. Long, J., Luo, Y., Zhu, X., Luo, E., Huang, M.: Computation offloading through mobile vehicles in IoT-edge-cloud network. EURASIP J. Wirel. Commun. Netw. **2020**(1), 244 (2020). https://doi.org/10.1186/s13638-020-01848-5
10. Atlam, H.F., Walters, R.J., Wills, G.B.: Fog computing and the Internet of Things: a review. Big Data Cogn. Comput. **2**(2), 10 (2018). https://doi.org/10.3390/bdcc2020010
11. Khan, W.Z., Ahmed, E., Hakak, S., Yaqoob, I., Ahmed, A.: Edge computing: a survey. Future Gener. Comput. Syst. **97**, 219–235 (2019). https://doi.org/10.1016/j.future.2019.02.050
12. Li, S., Xu, L.D., Zhao, S.: 5G Internet of Things: a survey. J. Ind. Inf. Integr. **10**, 1–9 (2018). https://doi.org/10.1016/j.jii.2018.01.005
13. Yang, M., Ma, Y., Liu, Z., Cai, H., Hu, X., Hu, B.: Undisturbed mental state assessment in the 5G era: a case study of depression detection based on facial expressions. IEEE Wirel. Commun. **28**(3), 46–53 (2021)
14. Ulah, A., Aznaoui, H., Batur Şahin, C., Sadie, M., Dinler, O.: Cloud computing and 5G challenges and open issues. Int. J. Adv. Appl. Sci. (2022). https://doi.org/10.11591/ijaas.v11.i3.pp187-193
15. Khanh, Q.V., Hoai, N.V., Manh, L.D., Le, A.N., Jeon, G.: Wireless communication technologies for IoT in 5G: vision, applications, and challenges. Wirel. Commun. Mob. Comput. **2022**, 1–12 (2022)
16. Chettri, L., Bera, R.: A comprehensive survey on Internet of Things (IoT) toward 5G wireless systems. IEEE Internet Things J. **7**(1), 16–32 (2020). https://doi.org/10.1109/JIOT.2019.2948888
17. Wang, Q., Guo, S., Liu, J., Yang, Y.: Energy-efficient computation offloading and resource allocation for delay-sensitive mobile edge computing. Sustain. Comput. Inform. Syst. **21**, 154–164 (2019). https://doi.org/10.1016/j.suscom.2019.01.007
18. You, C., Huang, K., Chae, H., Kim, B.-H.: Energy-efficient resource allocation for mobile-edge computation offloading. IEEE Trans. Wirel. Commun. **16**(3), 1397–1411 (2016). https://doi.org/10.1109/TWC.2016.2633522
19. Gai, K., Qiu, M., Zhao, H., Tao, L., Zong, Z.: Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing. J. Netw. Comput. Appl. **59**, 46–54 (2016). https://doi.org/10.1016/j.jnca.2015.05.016
20. Gai, K., Qiu, M., Zhao, H.: Energy-aware task assignment for mobile cyber-enabled applications in heterogeneous cloud computing. J. Parallel Distrib. Comput. **111**, 126–135 (2018). https://doi.org/10.1016/j.jpdc.2017.08.001
21. Kollu, A., Vadlamudi, S.: Computational intelligence techniques for energy management in cloud data centers. In: Proceeding of First Doctoral Symposium on Natural Computing Research: DSNCR 2020, pp. 185–193. Springer (2021). https://doi.org/10.1007/978-981-33-4073-2_19
22. Fernando, N., Loke, S.W., Rahayu, W.: Mobile cloud computing: a survey. Future Gener. Comput. Syst. **29**(1), 84–106 (2013). https://doi.org/10.1016/j.future.2012.05.023
23. Ahmed, E., Gani, A., Sookhak, M., Ab Hamid, S.H., Xia, F.: Application optimization in mobile cloud computing: motivation, taxonomies, and open challenges. J. Netw. Comput. Appl. **52**, 52–68 (2015). https://doi.org/10.1016/j.jnca.2015.02.003
24. Mishra, S.K., Sahoo, B., Parida, P.P.: Load balancing in cloud computing: a big picture. J. King Saud Univ. Comput. Inf. Sci. **32**(2), 149–158 (2020). https://doi.org/10.1016/j.jksuci.2018.01.003
25. Akherfi, K., Gerndt, M., Harroud, H.: Mobile cloud computing for computation offloading: issues and challenges. Appl. Comput. Inform. **14**(1), 1–16 (2018). https://doi.org/10.1016/j.aci.2016.11.002
26. Varghese, B., Buyya, R.: Next generation cloud computing: new trends and research directions. Future Gener. Comput. Syst. **79**, 849–861 (2018). https://doi.org/10.1016/j.future.2017.09.020
27. Khan, A.N., Kiah, M.M., Khan, S.U., Madani, S.A.: Towards secure mobile cloud computing: a survey. Future Gener. Comput. Syst. **29**(5), 1278–1299 (2013). https://doi.org/10.1016/j.future.2012.08.003
28. Li, W., Wu, J., Cao, J., Chen, N., Zhang, Q., Buyya, R.: Blockchain-based trust management in cloud computing systems: a taxonomy, review and future directions. J. Cloud Comput. **10**(1), 35 (2021). https://doi.org/10.1186/s13677-021-00247-5
29. Shi, W., Dustdar, S.: The promise of edge computing. Computer **49**(5), 78–81 (2016). https://doi.org/10.1109/MC.2016.145
30. Carvalho, G., Cabral, B., Pereira, V., Bernardino, J.: Edge computing: current trends, research challenges and future directions. Computing **103**, 993–1023 (2021). https://doi.org/10.1007/s00607-020-00896-5
31. Laroui, M., Nour, B., Moungla, H., Cherif, M.A., Afifi, H., Guizani, M.: Edge and fog computing for IoT: a survey on current research activities and future directions. Comput. Commun. **180**, 210–231 (2021). https://doi.org/10.1016/j.comcom.2021.09.003
32. Alwarafy, A., Al-Thelaya, K.A., Abdallah, M., Schneider, J., Hamdi, M.: A survey on security and privacy issues in edge-computing-assisted Internet of Things. IEEE Internet Things J. **8**(6), 4004–4022 (2020). https://doi.org/10.1109/JIOT.2020.3015432
33. Witanto, J.N., Lim, H., Atiquzzaman, M.: Adaptive selection of dynamic VM consolidation algorithm using neural network for cloud resource management. Future Gener. Comput. Syst. **87**, 35–42 (2018). https://doi.org/10.1016/j.future.2018.04.075
34. Nawrocki, P., Reszelewski, W.: Resource usage optimization in mobile cloud computing. Comput. Commun. **99**, 1–12 (2017). https://doi.org/10.1016/j.comcom.2016.12.009
35. Mohiuddin, K., Islam, A., Islam, M.A., Khaleel, M., Shahwar, S., Khan, S.A., Yasmin, S., Hussain, R.: Component-centric mobile cloud architecture performance evaluation: an analytical approach for unified models and component compatibility with next generation evolving technologies. Mob. Netw. Appl. (2022). https://doi.org/10.1007/s11036-022-01933-7
36. ...Maman, M., Calvanese-Strinati, E., Dinh, L.N., Haustein, T., Keusgen, W., Wittig, S., Schmieder, M., Barbarossa, S.,

Merluzzi, M., Costanzo, F., Sardellitti, S., Klessig, H., Kendre, S.V., Munaretto, D., Centenaro, M., di Pietro, N., Liang, S.-P., Chih, K.-Y., Luo, J.S.-J., Kao, L.-C., Huang, J.-C., Huang, J.-S., Wang, T.-Y.: Beyond private 5G networks: applications, architectures, operator models and technological enablers. EURASIP J. Wirel. Commun. Netw. **2021**(1), 195 (2021). https://doi.org/10.1186/s13638-021-02067-2

37. Mach, P., Becvar, Z.: Mobile edge computing: a survey on architecture and computation offloading. IEEE Commun. Surv. Tutor. **19**(3), 1628–1656 (2017). https://doi.org/10.1109/COMST.2017.2682318
38. ETSI, E.: Mobile Edge Computing (MEC); Framework and Reference Architecture (2016). http://www.etsi.org/deliver/etsi_gs/MEC/001_{0}99/003/01.01.01_{6}0/gs_MEC003v010101p.pdf. Accessed 4 Mar 2023
39. Roman, R., Lopez, J., Mambo, M.: Mobile edge computing, fog et al.: a survey and analysis of security threats and challenges. Future Gener. Comput. Syst. **78**, 680–698 (2018). https://doi.org/10.1016/j.future.2016.11.009
40. Li, B., Pei, Y., Wu, H., Shen, B.: Heuristics to allocate high-performance cloudlets for computation offloading in mobile ad hoc clouds. J. Supercomput. **71**, 3009–3036 (2015). https://doi.org/10.1007/s11227-015-1425-9
41. Powers, N., Alling, A., Osolinsky, K., Soyata, T., Zhu, M., Wang, H., Ba, H., Heinzelman, W., Shi, J., Kwon, M.: The cloudlet accelerator: Bringing mobile-cloud face recognition into real-time. In: 2015 IEEE Globecom Workshops (GC Wkshps), pp. 1–7. IEEE (2015). https://doi.org/10.1109/GLOCOMW.2015.7414055
42. Raei, H., Yazdani, N.: Performability Analysis of Cloudlet in Mobile Cloud Computing. Elsevier, New York (2017). https://doi.org/10.1016/j.ins.2017.01.030
43. Satyanarayanan, M., Bahl, P., Caceres, R., Davies, N.: The case for VM-based cloudlets in mobile computing. IEEE Pervasive Comput. **8**(4), 14–23 (2009). https://doi.org/10.1109/MPRV.2009.82
44. Jararweh, Y., Tawalbeh, L., Ababneh, F., Khreishah, A., Dosari, F.: Scalable cloudlet-based mobile computing model. Procedia Comput. Sci. **34**, 434–441 (2014). https://doi.org/10.1016/j.procs.2014.07.051
45. Xu, Z., Liang, W., Xu, W., Jia, M., Guo, S.: Capacitated cloudlet placements in Wireless Metropolitan Area Networks. In: 2015 IEEE 40th Conference on Local Computer Networks (LCN), pp. 570–578 (2015). https://doi.org/10.1109/LCN.2015.7366372
46. Marín-Tordera, E., Masip-Bruin, X., García-Almiñana, J., Jukan, A., Ren, G.-J., Zhu, J.: Do we all really know what a fog node is? Current trends towards an open definition. Comput. Commun. **109**, 117–130 (2017). https://doi.org/10.1016/j.comcom.2017.05.013
47. Deng, R., Lu, R., Lai, C., Luan, T.H., Liang, H.: Optimal workload allocation in fog–cloud computing toward balanced delay and power consumption. IEEE Internet Things J. **3**(6), 1171–1181 (2016). https://doi.org/10.1109/JIOT.2016.2565516
48. Ren, J., Zhang, D., He, S., Zhang, Y., Li, T.: A survey on end-edge-cloud orchestrated network computing paradigms: transparent computing, mobile edge computing, fog computing, and cloudlet. ACM Comput. Surv. **52**(6), 1–36 (2019). https://doi.org/10.1145/3362031
49. Zhang, P., Zhou, M., Fortino, G.: Security and trust issues in Fog computing: a survey. Future Gener. Comput. Syst. **88**, 16–27 (2018). https://doi.org/10.1016/j.future.2018.05.008
50. Yaqoob, I., Ahmed, E., Gani, A., Mokhtar, S., Imran, M., Guizani, S.: Mobile ad hoc cloud: a survey. Wirel. Commun. Mob. Comput. **16**(16), 2572–2589 (2016). https://doi.org/10.1002/wcm.2709
51. Mora, H., Colom, J.F., Gil, D., Jimeno-Morenilla, A.: Distributed computational model for shared processing on Cyber–Physical System environments. Comput. Commun. **111**, 68–83 (2017). https://doi.org/10.1016/j.comcom.2017.07.009
52. Jhaveri, R.H., Patel, N.M., Zhong, Y., Sangaiah, A.K.: Sensitivity analysis of an attack-pattern discovery based trusted routing scheme for mobile ad-hoc networks in Industrial IoT. IEEE Access **6**, 20085–20103 (2018). https://doi.org/10.1109/ACCESS.2018.2822945
53. Kuwata, Y., Teo, J., Fiore, G., Karaman, S., Frazzoli, E., How, J.P.: Real-time motion planning with applications to autonomous urban driving. IEEE Trans. Control Syst. Technol. **17**(5), 1105–1118 (2009). https://doi.org/10.1109/TCST.2008.2012116
54. Farris, I., Militano, L., Nitti, M., Atzori, L., Iera, A.: MIFaaS: A mobile-IoT-federation-as-a-service model for dynamic cooperation of IoT cloud providers. Elsevier (2017). https://doi.org/10.1016/j.trc.2015.09.011
55. Kim, D.-S., Kwon, J.: Moving object detection on a vehicle mounted back-up camera. Sensors **16**(1), 23 (2016). https://doi.org/10.3390/s16010023
56. Huang, C.-M., Chiang, M.-S., Dao, D.-T., Su, W.-L., Xu, S., Zhou, H.: V2V data offloading for cellular network based on the software defined network (SDN) inside mobile edge computing (MEC) architecture. IEEE Access **6**, 17741–17755 (2018). https://doi.org/10.1109/ACCESS.2018.2820679
57. Ge, X., Tu, S., Mao, G., Wang, C.-X., Han, T.: 5G ultra-dense cellular networks. IEEE Wirel. Commun. **23**(1), 72–79 (2016). https://doi.org/10.1109/MWC.2016.7422408
58. Zhou, B., Buyya, R.: Augmentation techniques for mobile cloud computing: a taxonomy, survey, and future directions. ACM Comput. Surv. (CSUR) **51**(1), 1–38 (2018). https://doi.org/10.1145/3152397
59. Okegbile, S.D., Maharaj, B.T., Alfa, A.S.: A multi-user tasks offloading scheme for integrated edge–fog–cloud computing environments. IEEE Trans. Veh. Technol. **71**(7), 7487–7502 (2022). https://doi.org/10.1109/TVT.2022.3167892
60. Zhou, B., Dastjerdi, A.V., Calheiros, R.N., Buyya, R.: An online algorithm for task offloading in heterogeneous mobile clouds. ACM Trans. Internet Technol. **18**(2), 23–12325 (2018). https://doi.org/10.1145/3122981
61. Whaiduzzaman, M., Naveed, A., Gani, A.: MobiCoRE: mobile device based cloudlet resource enhancement for optimal task response. IEEE Trans. Serv. Comput. **11**(1), 144–154 (2018). https://doi.org/10.1109/TSC.2016.2564407
62. Kaya, M., Koçyiğit, A., Eren, P.E.: An adaptive mobile cloud computing framework using a call graph based model. J. Netw. Comput. Appl. **65**, 12–35 (2016). https://doi.org/10.1016/j.jnca.2016.02.013
63. da Mata, S.H., Guardieiro, P.R.: Resource allocation for the LTE uplink based on Genetic Algorithms in mixed traffic environments. Comput. Commun. **107**, 125–137 (2017). https://doi.org/10.1016/j.comcom.2017.04.004

Higinio Mora received the B.S. Degree in Computer Science Engineering and the B.S. Degree in Business Studies from the University of Alicante, Spain, in 1996 and 1997, respectively, and the Ph.D. Degree in Computer Science from the University of Alicante in 2003. Since 2002, he is a Member of the Faculty of the Computer Technology and Computation Department, University of Alicante, where he is currently an Associate Professor and a Researcher with the Specialized Processors Architecture Laboratory. His research interests include Computer Modeling, Computer Architectures, High-Performance Computing, Embedded Systems, the Internet of Things, and Cloud Computing paradigm. He has participated in many conferences and most of his work has been published in international journals and conferences, with more than 50 published articles.

Francisco A. Pujol received the B.S. Degree in Telecommunications Engineering from the Polytechnic University of Valencia, Spain, in 1998, and a Ph.D. Degree in Computer Science from the University of Alicante, Spain, in 2001. He is currently an Associate Professor with the Department of Computer Technology and Computation, University of Alicante. He was a Visiting Lecturer at Cardiff University (2004). His research interests focus on Robotics, Machine Learning, Face Recognition, Computer Vision, and Computer Parallel Architectures, on which he has published more than 100 technical journals and conference papers.



Tamai Ramírez received the B.S. Degree in Robotics Engineering from the University of Alicante, Spain, in 2022. He is currently a Researcher in the Department of Computer Technology and Computation, University of Alicante. His research interest focus on Cloud Computing, Artificial Intelligence, Internet of Things, Machine Learning and Computer Vision. He is finishing M.S. Degree in Artificial Intelligence from the Valencia International University, Spain, in July 2023.

Antonio Jimeno-Morenilla currently works at the Department of Computer Technology and Computation, University of Alicante. He has been researching into CAD/CAM and HPC applied to the footwear sector for over 20 years. Furthermore, he has led the University of Alicante's Research Group UniCAD since 2008. Its results have appeared in more than 70 publications in high impact factor journals (JCR). As regards technology transfer, he has taken part in the development of top-class software, with more than 2000 operating licences distributed around the world. He has also been involved in approximately 20 research projects and agreements, in 11 of which he was Principal Investigator.

Julian Szymański received the Ph.D. Degree. He is currently with the Department of Computer Architecture, Faculty of Electronics, Telecommunications and Informatics, Gdansk University of Technology. He addresses the problems of knowledge representation, methods of lexical knowledge acquisition, and linguistic data utilization. His research results mainly find applications in web search engines and systems of automated text categorization. He manages research projects for processing the information in Wikipedia. His main goal is to extract and structuralize the textual knowledge and construct interfaces capable of communicating in natural language. Besides his research in information retrieval domains, he involved with projects related to cognitive science. His mainstream of this research is building semantic memory models that improve natural language processing. He also involved in the area of analysing human emotions, by mining EEG signals. His results are implemented in the form of brain-machine interfaces for disabled people. He also works on analysing data with the IoT domain. He implements a set of sensors for an acquisition of data from beehives that allows one to predict apiary development and detect abnormalities. He served for many international conferences, including the International Conference on Networked Digital Technologies (NDT), the International Symposium on Innovations in Intelligent Systems and Applications (INISTA), and the International Conference on Parallel and Distributed Systems (ICPDS). His research interest includes application of data-mining methods to natural language processing (NLP). He was a Reviewer for international journals and conference proceedings.