HAL
open science

Metrology for 5G edge networks (MEC). Leveraging mobile devices beyond the edge toward task offloading

Sadia Khizar

To cite this version:

Sadia Khizar. Metrology for 5G edge networks (MEC). Leveraging mobile devices beyond the edge toward task offloading. Mobile Computing. Sorbonne Université, 2022. English. NNT : 2022SORUS069 . tel-03715393

HAL Id: tel-03715393

https://theses.hal.science/tel-03715393v1

Submitted on 6 Jul 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

PhD THESIS OF
SORBONNE UNIVERSITÉ

Speciality

Computer Science

École doctorale Informatique, Télécommunications et Électronique de Paris
Thesis conducted at LIP6

Presented by

Sadia KHIZAR

Submitted for the degree of

DOCTOR of SORBONNE UNIVERSITÉ

PhD Thesis :

Metrology for 5G edge networks (MEC)
Leveraging mobile devices beyond the edge toward task offloading

to the committee composed of:

Nadjib Ait Saadi, Prof. Université Versailles Saint-Quentin	Reviewer
André-Luc Beylot, Prof. ENSEEIHT	Reviewer
Anne Fladenmuller, Prof. Sorbonne Université	Examiner
Thi-Mai-Trang Nguyen, Assoc. Pr Sorbonne Université	Examiner
Marcelo Dias de Amorim, DR CNRS	Co-Advisor
Vania Conan, DR Thales	Co-Advisor

SORBONNE UNIVERSITÉ
CRÉATEURS DE FUTURS DEPUIS 1257

This page intentionally left blank

THÈSE DE DOCTORAT DE
SORBONNE UNIVERSITÉ

Spécialité

Informatique

École doctorale Informatique, Télécommunications et Électronique de Paris
Thèse effectuée au LIP6

Présentée par

Sadia KHIZAR

Pour obtenir le grade de

DOCTEURE de SORBONNE UNIVERSITÉ

Sujet de la thèse :

Métrologie des réseaux de bordure 5G (MEC)

Tirer profit des dispositifs mobiles au-delà de la périphérie afin de délester des tâches

devant le jury composé de :

Nadjib Ait Saadi, Pr Université Versailles Saint-Quentin	Rapporteur
André-Luc Beylot, Pr ENSEEIHT	Rapporteur
Anne Fladenmuller, Pr Sorbonne Université	Examinatrice
Thi-Mai-Trang Nguyen, MCF Sorbonne Université	Examinatrice
Marcelo Dias de Amorim, DR CNRS	Co-Directeur
Vania Conan, DR Thales	Co-Directeur

SORBONNE UNIVERSITÉ
CRÉATEURS DE FUTURS DEPUIS 1257

I dedicate my work to everyone who has contributed to becoming the person I am today. Especially my father, who dreamed of seeing us shine, and my mother, who supported this dream for over 30 years.

Remerciements

Cette thèse restera à jamais gravée dans mon cœur et ma mémoire. À l'écriture de ces remerciements un sentiment étrange est présent, difficile à décrire. C'est un mélange de bonheur, de fierté, de surprise, de mélancolie. Eh oui ! Tous ces sentiments m'habitent. La surprise d'en être arrivée là. Le bonheur et la fierté du chemin parcouru. Et bien sûr, un pincement au cœur de se dire qu'une page se tourne, de se remémorer toutes les étapes de cette aventure. Ainsi, cette thèse m'a permis de m'enrichir d'un point de vue personnel et scientifique, en m'apportant de nouveaux regards sur le monde qui m'entoure. Cette aventure n'a été possible que grâce à une multitude de mains tendues, qui m'ont soutenue, donné, transmis. Une thèse ne se fait pas toute seule, c'est pourquoi je tiens à remercier et à montrer par ces quelques mots ma plus profonde gratitude à toutes ces personnes.

Tout d'abord, une thèse ne serait rien sans ses encadrants. Ainsi, en tout premier lieu, j'adresse mes sincères remerciements à mes deux encadrants, Marcelo Dias de Amorim et Vania Conan, pour avoir accepté d'encadrer ma thèse. Je suis très reconnaissante pour les nombreux enseignements qu'ils m'ont transmis, aussi bien sur le plan technique que pour les travaux de synthèse et de rédaction, pour leurs précieux conseils et leur disponibilité tout au long de cette période. Je remercie « M.Marcelo » de m'avoir transmis le sens de la rigueur mais surtout du détail, tout en allant droit au but. Merci pour vos anecdotes qui avaient, je pense, toujours un but d'enseignement avant tout. Je remercie Vania Conan non seulement de m'avoir accueilli chaleureusement au sein de CNS, mais également d'avoir suivi avec intérêt mes travaux tout au long de ces trois années. Merci de m'avoir encouragée et d'avoir transformé mes peurs et mes craintes. Je réalise aujourd'hui que « le petit morceau par petit morceau » a finalement abouti à la réalisation de la thèse, comme prévu. La qualité de votre encadrement, la confiance que vous m'avez accordée et la passion que vous m'avez transmise ont permis la réalisation de cette thèse. Cette belle expérience m'a permis de mûrir scientifiquement, mais aussi humainement, et ce, notamment grâce à vous. Je vous remercie également pour tous les moments de joie passés avec vous, que ce soit en réunion ou au détour d'un café, et qui ont égayé ces trois années

de thèse.

Je voudrais aussi remercier André-Luc Beylot et Nadjib Ait Saadi qui ont bien voulu commenter mon manuscrit de thèse. Je suis consciente qu'ils ont passé beaucoup de temps à le lire et à rédiger leur rapport. Leurs remarques m'ont permis d'améliorer la qualité du manuscrit et de la présentation. Je tiens aussi à remercier Anne Fladenmuller et Thi-Mai-Trang Ngyuen pour avoir accepté de faire partie de mon jury et d'évaluer les travaux que je propose.

Cette aventure n'aurait pas été la même sans mes collègues du laboratoire CNS de Thales. Ces trois années passées à vos côtés ont été très enrichissantes, j'y ai trouvé un cadre très amical, bienveillant et vivant où j'ai eu le plaisir de travailler et d'échanger avec vous tous. Merci pour les nombreux pots, les moments de rire, et vos conseils tout au long de ces trois années que je n'oublierai pas. Je remercie tous mes collègues et anciens collègues qui ont rendu l'ambiance de travail très agréable : Adrien, Agathe, Alan, Azharia, Bruno, Clément, Damien, Ehsan, Erwan, Farid, Filippo, Gaëtan, Geoffroy, Hicham, Jean-Marc, Kévin, Marie-Noëlle, Mathieu, Raphaël, Simon, Vania. Merci pour votre soutien et ces longues discussions qui m'ont tant apporté au cours de ces trois ans.

Je garde aussi un très bon souvenir des moments passés au LIP6 un jour par semaine. J'ai beaucoup apprécié l'esprit d'équipe entre les doctorants et les divers échanges que nous avons eus. Je remercie Maria pour son accueil au sein de l'équipe NPA et les doctorants et anciens doctorants: Burim, Gewu, Imran, Kévin, Matthieu, Maxime, Yakolley. Je suis aussi reconnaissante envers Sabine qui a toujours su m'aider avec gentillesse dans la gestion des différentes procédures Lip6 tout au long de ma thèse.

L'aboutissement de ces trois années de thèse est aussi l'occasion pour moi de remercier toutes les personnes qui me sont chères, et qui ont contribué à ce travail à leur manière: Coralie, Elisa, Hamza G, Hamza K, Ibthial, Jean-Guy, Maha, Mariam, Meryam, Nabia, Muzzamil, Nicolas, Salim, Yacine, Wander. Un remerciement particulier à mon ami pilier: Fred alias BIDOU, merci d'être une source de stabilité et de joie dans ma vie !!

Finalement, je tiens à remercier toute ma famille qui est une source de soutien indéfectible. Plus particulièrement, je tiens à remercier mes parents Hayat et Shakila. Merci d'avoir toujours été là pour moi, merci de croire en moi plus que je ne sais le faire, merci pour toutes les valeurs que vous m'avez transmises, rien de tout cela n'aurait été possible sans vous. Je tiens également à remercier particulièrement mon frère et mes sœurs: Junaid et Saba, Saima. Depuis l'enfance, vous avez toujours été des compagnons de route et surtout une source de bonheur absolue. Un merci spécial pour une personne spéciale dans ma vie, merci à mon cher Jamil alias KIYANI, qui m'a soutenu pendant ces trois dernières années. Je sais que ce n'était pas simple. Tu as su me comprendre, m'accepter.

Résumé

La bataille perpétuelle consiste à améliorer les performances en réduisant la latence et en approvisionnant des ressources tout en garantissant une qualité de service élevée aux utilisateurs finaux.

Selon les prévisions de Cisco, il y aura 29,3 milliards d'appareils en réseau d'ici 2023. L'explosion des appareils mobiles s'accompagne d'une quantité tout aussi massive de données à traiter. Plusieurs technologies sont présentes pour répondre à ces exigences, notamment le *cloud computing* et l'*edge computing*. Nous nous intéressons à l'une de ces solutions, qui suscite depuis peu un intérêt croissant de la part de la communauté des chercheurs : le *multi-access edge computing* (MEC).

La tendance est de considérer le MEC comme une solution de calcul incontournable. Le MEC permet d'exécuter des applications sensibles aux délais et contextuelle à proximité des utilisateurs finaux, tout en réduisant l'utilisation de la bande passante et le calcul au cœur du réseau. Néanmoins, pour faire face à la croissance des appareils mobiles, un déploiement à grande échelle de la technologie MEC est nécessaire, mais cela exige des investissements importants dans le réseau d'accès radio.

L'omniprésence des dispositifs mobiles équipés d'une connectivité Internet et de systèmes de positionnement, qu'il s'agisse de véhicules, de téléphones mobiles, de smartwatches ou de bracelets de fitness, cela nous amène à les considérer comme une ressource précieuse à exploiter. Dans cette thèse, nous abordons l'utilisation des dispositifs mobiles sous un nouvel angle. Nous considérons l'extension de la capacité du MEC en utilisant les ressources disponibles des dispositifs mobiles *au-delà de la bordure* du réseau d'infrastructure. L'objectif est de maximiser l'utilisation des appareils mobiles et de tirer parti de leurs ressources inexploitées pour traiter les calculs pour le compte du MEC d'une manière distribuée. Pour pouvoir s'appuyer sur des nœuds mobiles, il est fondamental que l'orchestrateur MEC soit capable de connaître son environnement d'exploitation. Par conséquent, le MEC a besoin d'estimer s'il dispose de suffisamment de ressources de calcul, de stockage et de batterie, mais également de déterminer pour *combien de temps*.

Dans la première partie de la thèse, nous nous sommes intéressés à la disponibilité temporelle des ressources au-delà du bord. Nous avons choisi d'étudier la co-localisation des terminaux et d'analyser leurs *persistance* dans une cellule. Dans la deuxième partie de la thèse, nous nous intéressons à l'allocation des tâches. Nous mettons l'accent sur l'aspect spatio-temporel en quantifiant les ressources qu'une cellule peut fournir pour effectuer une tâche MEC. Nous estimons le *potentiel* de tâches de calcul effectuées par les nœuds en fonction du temps de présence cumulé dans une cellule donnée et d'un délai d'achèvement donné.

Nous avons exploré de manière spatiale et temporelle la disponibilité potentielle des ressources informatiques au-delà de la bordure dans des zones urbaines. Les résultats donnent un aperçu des possibilités de décharger les tâches de calcul sur les nœuds mobiles. De plus, ils permettent de connaître les endroits où il est judicieux de délester les tâches ainsi que la durée des tâches pouvant être délestées. En complément, le MEC peut avoir une garantie spécifique de disposer d'un nombre suffisant de dispositifs mobiles pour effectuer une tâche. Enfin, les résultats donnent un aperçu des tâches supplémentaires réalisables en fonction de la tolérance face au délai d'exécution que le MEC est prêt à accepter.

Abstract

The perpetual battle is to improve performance by reducing latency and provisioning resources while ensuring a high quality of service for end-users.

According to Cisco's forecast, there will be 29.3 billion networked devices by 2023. The outburst of mobile devices comes with an equally massive amount of data to process. Several technologies are present to tackle these requirements, including the *cloud* and the *edge computing*. We turn our attention to one of these solutions, recently attracting increasing interest from the research community: the *multi-access edge computing* (MEC).

The trend is to consider MEC as a compelling computing facility. MEC allows executing delay-sensitive and context-aware applications close to the end-users while alleviating backhaul utilization and computation at the core network. Nevertheless, accommodating the growth of mobile devices with a large-scale deployment of MEC requires significant investments in the radio access network.

The pervasiveness of mobile devices equipped with internet connectivity and positioning systems, ranging from vehicles, mobile phones to smartwatches and fitness bracelets, leads us to regard them as a valuable resource to leverage. In this thesis, we tackle the use of mobile devices from a new perspective. We consider the extension of the capacity of the MEC by using the available resources of mobile devices *beyond the edge* of the infrastructure network. The goal is to maximize the use of mobile devices and leverage their untapped resources to process computation on behalf of the MEC in a distributed way. It is fundamental for the MEC orchestrator to be aware of its operating environment to rely on mobile nodes. Therefore, MEC needs to estimate if they have enough computing, storage, and battery resources, but also for *how long*.

In the first part of the thesis, we have focused on the temporal availability of beyond-the-edge resources. We chose to investigate the co-location of terminals and analyze their *persistence* in a cell. In the second part of the thesis, we turn our attention to task allocation. We shift the focus on the spatio-temporal aspect by quantifying the resources that a cell can provide to perform a MEC task. We estimate the *potential* amount of

computational tasks performed by nodes based on the cumulative presence time in a given cell and a given completion delay.

We have explored spatially and temporally the potential availability of computing resources beyond the edge in urban areas. Results provide insight into the possibilities of offloading computing tasks on mobile nodes. Furthermore, it allows knowing the locations where it is advisable to offload tasks and the time duration of tasks offloadable. In addition, the MEC may have a specific guarantee of having a sufficient number of mobile devices to perform a task. Finally, the results give an overview of additional tasks feasible based on the delay completion tolerance that the MEC is willing to accept.

Contents

1 Introduction **19**

1.1 Context and motivations . . . 19

1.2 Mobile devices: An extra computing resource . . . 20

1.3 Approach . . . 21

1.4 Exploitation of real-world datasets . . . 22

1.5 Challenges and contributions . . . 24

1.6 Thesis outline . . . 27

2 Positioning and Related Works **29**

2.1 Mobile Cloud Computing . . . 30

2.2 Multi-access edge computing . . . 31

2.3 Offloading to the MEC . . . 35

2.4 Discussion . . . 37

2.5 Conclusion . . . 39

3 Persistence of Vehicular-Augmented Mobile Edges **41**

3.1 Persistence of sets of nodes . . . 43

3.1.1 Vehicular datasets . . . 44

3.1.2 Graphical view of persistence . . . 47

3.2 Detailled analyses . . . 48

3.2.1 Persistence in Rome (Taxis) . . . 48

3.2.2 Persistence in Rio (Buses) . . . 50

3.3 Varying the reduction threshold . . . 52

3.3.1 Influence of the reduction threshold in Rome . . . 52

3.3.2 Influence of the reduction threshold in Rio . . . 54

3.4 Impact of cell size on the persistence . . . 55

3.4.1 Average persistence evolution in Rome using 1 km^2 cells . . . 56

3.4.2 Average persistence evolution in Rio using 1 km^2 cells 56
3.4.3 Variation of cells size in Rome 59
3.4.4 Variation of cells size in Rio 60
3.5 Analysis of the ability to accomplish task 61
3.5.1 Case 1: Percentage of successful tasks when at least one node is present in the set 63
3.5.2 Case 2: Percentage of successful tasks when sets of nodes stay together as formed. 63
3.6 Conclusion 64

4 Offloading computing tasks beyond the edge in a cell 67
4.1 Offloading strategy and assumptions 69
4.1.1 Task allocation process 70
4.1.2 Cumulative presence time and completion delay 71
4.1.3 Formalization 71
4.2 Pedestrian dataset 72
4.2.1 Number of users in cells 73
4.2.2 Average frequency of nodes return 74
4.2.3 Average presence time of node in a cell 75
4.3 Varying the completion delay 76
4.4 Capacities provided by a node in a cell 79
4.4.1 Case 1: Average number of tasks with a zero completion tolerance delay 79
4.4.2 Case 2: Average number of tasks with a 24-hour completion tolerance delay 80
4.4.3 The gain derived from the completion delay 83
4.5 Conclusion 84

5 Conclusion and futures perspectives 85
5.1 Conclusion 85
5.2 Future perspectives 86

Appendix A Résumé détaillé en Français 89
A.1 Introduction 89
A.2 Les appareils mobiles : une ressource informatique supplémentaire 91
A.3 Challenges 92

A.3.1 Contribution 1 : L'exploitation des dispositifs mobiles dans un scénario véhiculaire 94
A.3.2 Contribution 2 : L'exploitation des dispositifs mobiles dans un scénario piéton 95
A.4 Conclusion 96
A.5 Perspectives 97

Bibliography **98**

List of Figures

1.1 Hierarchical infrastructure. 20
1.2 Extension of the MEC beyond the edge. 22
1.3 Challenges to exploit mobile devices at beyond the edge. 23
1.4 Our journey to reflection on this topic. 27

2.1 MEC architecture enhancing applications and low latency. 32
2.2 Cloudlet. 33
2.3 From conventional architectures towards leveraging mobile devices resources. 38

3.1 Formation of sets and their evolution over time. 45
3.2 Simulation area of Rome (256 km^2). 46
3.3 Simulation area of Rio de Janeiro (256 km^2). 47
3.4 Persistence of a cell between 7:00 am and 8:00 am in a dense area of Rome. 48
3.5 Persistence of a cell between 7:00 and 8:00 am in a dense area of Rio. . . . 49
3.6 Downtown, dense area (Rome). 50
3.7 Periphery, sparse area (Rome). 51
3.8 Downtown, dense area (Rio). 52
3.9 Periphery, sparse area (Rio). 53
3.10 Average persistence of sets in Rome and their evolution over time for 1 km^2 cell size. 57
3.11 Average persistence of sets in Rio and their evolution over time for 1 km^2 cell size. 58
3.12 Cell size variation in Rome from 6:00 am to 9:00 am. 60
3.13 Cell size variation in Rio from 4:00 am to 7:00 am. 62
3.14 Task well-executed for each dataset and scenario. 65

4.1 Offload computing tasks to mobile devices without losing the task processing when they leave the cell. 70

4.2 Illustration of the presence time and completion delay. 71
4.3 Number of user in cells during 24h. 72
4.4 Cell distribution of nodes number. 73
4.5 Average node return in cell. 75
4.6 Distribution of the average presence time of nodes in cells. 76
4.7 Impact of varying task completion delay. The plots show the number of task offloading opportunities, for the two cells, for tasks of duration ranging from 10 to 35 min, and with increasing completion delay (up to 120 min). 78
4.8 Average number of tasks with a zero completion tolerance delay. 79
4.9 Average number of tasks with a 24-hour completion tolerance delay. 81
4.10 Additional tasks gain opportunities obtained by using completion delay. . . 82

A.1 Infrastructure hiérarchisée. 90
A.2 Extension du MEC au-delà du bordure. 92
A.3 Les défis à relever pour exploiter les appareils mobiles au-delà de la bordure. 93
A.4 Notre cheminement vers la réflexion sur cette thématique. 96

List of Tables

2.1 Difference between Fog computing and MEC. 34

3.1 Reduction Threshold Variation for Rome 54
3.2 Reduction Threshold Variation for Rio 55

—Start by doing what's necessary; then do what's possible; and suddenly you are doing the impossible—

Francis of Assisi

1 Introduction

This chapter provides the general motivation for this research, highlighting the key benefits and challenges of mobile devices exploitation. It also provides an introduction to the organization of the thesis along with its contributions.

1.1 Context and motivations

Moreover, the exponential growth of connected devices reflects the avalanche effect of the increasing volume of mobile and wireless traffic. According to Cisco's forecast, there will be 29.3 billion networked devices by 2023 [1], to which operators have to maintain a high quality of service.

To accommodate these requirements, academia and industry explore multiple technologies. The answer to where we may find improved availability of computing resources, together with the required dependability to provide smooth connectivity for wireless applications, inevitably revolves around the *cloud*. A cloud is a tried-and-true approach for extending the resource capabilities of end-devices with solid data center architectures. For example, the practice of offloading computing tasks toward the cloud is a well-known solution. However, the latency caused by the distance between the end device and the cloud infrastructure and the cost of traversing the backhaul network is no longer suit-

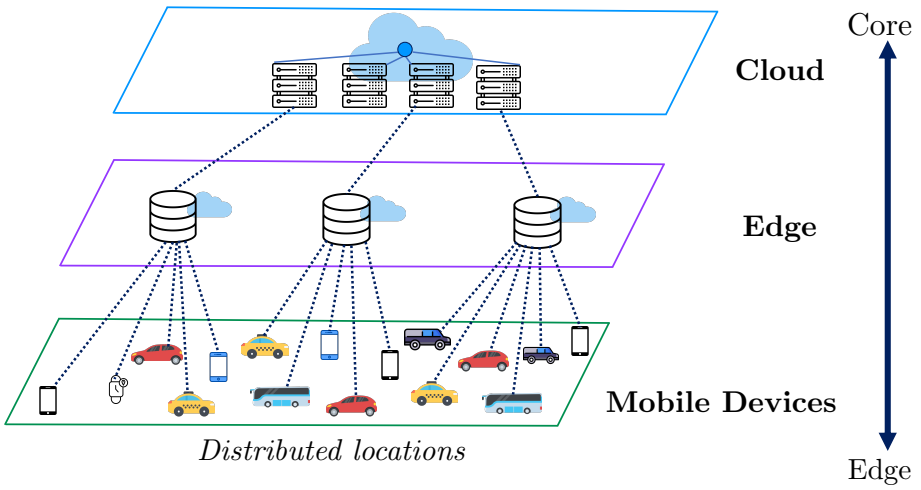


Figure 1.1: Hierarchical infrastructure.

able for resource-intensive applications. Hence, it leads to adopting alternative solutions. There is no doubt that these alternative solutions need to introduce a more distributed infrastructure that improves local efficiency by bringing cloud-like features closer to mobile devices.

In the era of the computing paradigm, edge computing has started to take an important place, especially the *multi-access edge computing* (MEC). MEC provides computing capacities closer to mobile users by shifting cloud services to the radio access network [2, 3, 4, 5, 6]. Thus, MEC solves cloud computing latency and traffic congestion problems that transmit large amounts of data outside the core network. MEC is an intermediary layer that exists between the cloud and mobile devices. Figure 1.1 depicts the construction of a three-hierarchical structure. MEC uses cloud computing to assist and improve the performance of end devices.

However, large-scale deployment and high costs make it difficult to reap the benefits of MEC [7]. The impetus of 5G and the costs of deploying MEC have prompted the pursuit of alternative methods to alleviate cellular network pressure during this thesis. Therefore, we turn our attention to mobile devices to address network load, latency issues, improve bandwidth, and provide computational resources.

1.2 Mobile devices: An extra computing resource

We consider the scenario where end-devices help the MEC run computing tasks. At first sight, the use of mobile devices to extend the capacity of the MEC seems counter-intuitive. Indeed, all technologies mentioned above intend to overcome the problems posed

by the compact size of mobile devices (i.e., limited battery and insufficient computing resources). Nevertheless, mobile devices have witnessed a remarkable evolution from small devices with limited resource capabilities to becoming essential accessories for everyday life. They have acquired more power and possess increased computing, storing, and sensing capacities as the years go by. In addition, they are becoming more affordable for consumers, making them easier to purchase. Above all, they are rarely used to their full capabilities and are often left idle [8, 9].

Given mobile devices' widespread use, it drives us to consider them as a valuable resource to leverage. Moreover, they are a pre-existing resource that does not generate any deployment cost for operators. Thus, we claim that mobile devices are a crucial element to exploit for operators, particularly to mitigate the massive amount of data to process and the cost that it generates. The motto *"unity makes strength"* leads us to assume that perhaps a device has insufficient capacity but a billion changes the vision. Therefore, we aim to take advantage of mobile devices and explore the possibility of using them to process distributed computations. A resulting question is thus *to what extent mobile devices provide computational resources.*

This thesis focus on a different approach; for the reader's convenience, we depict it in Figure 1.2. It consists of offloading computing tasks from the MEC servers to the mobile devices, thus performing computation *beyond the edge*. In that way, mobile devices serve as potential mobile workstations to assist in relieving the load from the network infrastructure. They can also improve MEC server availability. As a result, it may increase overall MEC performance. In this thesis, thus, we tackle the measurement of the available resources beyond the edge, enabling the MEC capacities extension and overall relief of network infrastructure. Such an approach is all the more promising given the plethora of mobile devices present beyond the edge. Thus, using untapped resources by running tasks appears promising. However, many challenges still need to be overcome to confirm its viability.

1.3 Approach

It is essential that the orchestrator MEC has a current and accurate view of what is happening beyond the edge. It needs to be aware of the dynamics of its environment or the potentially available resources at all levels. This awareness is the key to improving its management and the execution of its tasks. For example, a significant challenge to offload tasks from the MEC to mobiles devices lies in the complexity inherent to mobility. Indeed, users move with their devices when communicating at work, amusement parks, shopping

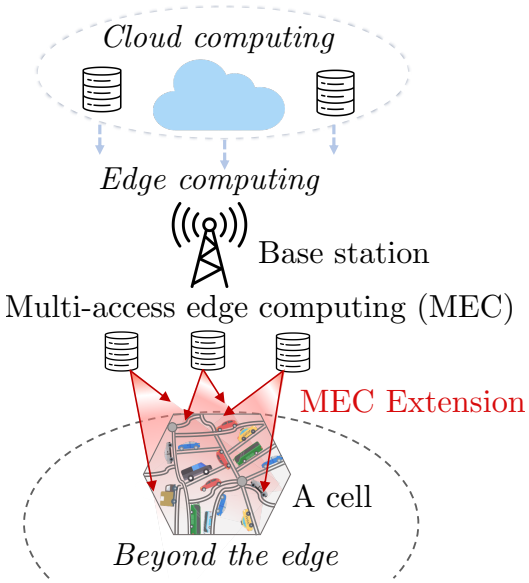


Figure 1.2: Extension of the MEC beyond the edge.

malls. Subsequently, mobility affects the potential resources available at mobile devices. Moreover, for the MEC to rely on terminals to run computing tasks, it is fundamental that the MEC orchestrator has an insight into the states of mobile devices at a given moment. For instance, it requires estimating if the mobile devices have enough computing, storage, and battery resources as depicted in Figure 1.3. Furthermore, it is evident that without these conditions, a MEC is incapable of offloading a task onto end-devices. *In this thesis, our primary focus is analyzing the mobility characteristics of mobile devices and check whether they can help the MEC relieve its load. To this end, we consider real-world datasets.*

1.4 Exploitation of real-world datasets

Mobile operators constantly monitor mobile traffic on their networks for troubleshooting, efficiency, and billing purposes. However, they have traditionally been very cautious about the disclosure of the collected data. Sociologists, epidemiologists, physicists, transportation, and telecommunication experts recognized a clear opportunity to bring their analyses to an unprecedented scale in the datasets collected by mobile network operators. Overall, they might improve understanding natural or technological phenomena occurring at large scales while keeping a high degree of information on each individual. The mindset of operators has shifted in recent years as they have grown more willing to share their data with the academic community. Such a change was made feasible by the bene-

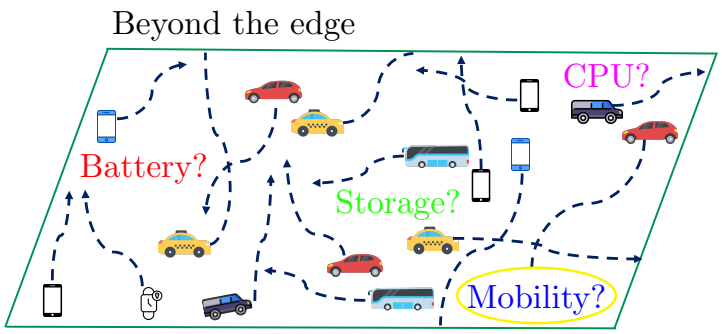


Figure 1.3: Challenges to exploit mobile devices at beyond the edge.

ficial work proving that mobile traffic data is an invaluable source for basic research that benefits operators themselves. Thus, the collaboration between academic research groups and network operators based on real-world mobile traffic data sets is thriving, resulting in increased research and publication. Moreover, the improving quality of datasets is a crucial indicator of the study field's success.

We can divide mobile traffic analysis studies into three main categories: social analysis, mobility analysis, network analysis [10]. Our study focuses essentially on mobility analysis to evaluate the possibility of leveraging mobile devices on behalf of the MEC computing tasks. Namely, there is not a net separation with the other categories; instead, they are almost interdependent. For example, indirectly, we get information about the social analysis, such as the demographic aspect or even the interactions between users.

Besides the main objectives of this thesis, the usage of datasets has made us aware of the importance and complexity of properly manipulating and interpreting them. Therefore, we have conducted statistical studies to describe and explore the data in-depth. During this thesis, we use two categories of data sources; two Global Positioning Systems (GPS) data from Crawdad and one data collected at core-edge network precisely at the RAN probe by a nationwide operational mobile network. These datasets capture spatio-temporal footprints of vehicular and pedestrian mobility with high spatial and temporal resolutions. In the vehicular case, we consider two datasets: (i) 320 taxi cabs in Rome, Italy [11] and (ii) 12,000 buses in Rio de Janeiro, Brazil [12]. In the pedestrian case, we collected a dataset a major European mobile operator's core network. It describes the mobile traffic generated by all subscribers in Paris [13]. These datasets are inherently different and allow us to understand better the possibilities that mobile devices offer in terms of potential offloading.

1.5 Challenges and contributions

This thesis introduces a new cutting-edge idea leveraging mobile devices to relieve the network infrastructure. Indeed, compared to the existing solution such as infrastructure offloading only (i.e., cloud, edge computing) [14, 15, 16, 17, 18], the access point-based offloading method [19, 20], the concept of offloading to a coordinator node near the end-users [21, 22] or in particular the D2D offloading mechanism [23, 24]; another adaptive approach has been proposed in this framework to untapped resources for mobile edge computing task execution. As mentioned earlier, given mobile devices' significant presence, our main interest turns around knowing how they would serve to extend the capabilities of the MEC. Nevertheless, leveraging mobile devices begins by understanding their environment. Thus, the turning point of our study is to consider the characterization of the resources beyond the edge. MEC has to estimate the volume and duration of available resources that it can use from mobile devices. Thus, the resulting challenges are:

Challenge 1: How often and how long mobile devices remain co-located, so that they can extend, collectively, the capacity of the MEC?

This question is crucial since it will allow us to assert whether we can utilize mobile devices. Moreover, it will avoid failure when the MEC is about to assign computing tasks to mobile devices. The heart of this question is the temporal availability of resources beyond the edge.

Challenge 2: How many tasks assigned by the MEC mobile devices can undertake?

This question provides insights into how valuable mobile devices are for MEC extension. So, we shift our attention to the potential allocation of tasks. We emphasize the spatial aspect by quantifying the resources a cell can provide to perform a MEC task.

In this context, the contributions of this thesis are:

Contribution 1: Leveraging mobile devices in a vehicular scenario

We investigate the computation offloading capacity that a set of vehicles in a cell can offer to the MEC. We call the size of this set the "potential" offloading capacity of the vehicular cloud. As vehicles are highly mobile, it is fundamental to investigate the dynamics of the

potential offloading capacity. Therefore, when the MEC is about to assign computing tasks to vehicles, it first needs to estimate the lifetime of vehicles composing the vehicular cloud. In short, the question we consider is: "given the set of vehicles in a cell at time t, what is the expected duration of this set before its size decreases to a given percentage of its initial size?". We refer to this duration as the *persistence* of the potential offloading capacity.

A straightforward way to estimate the persistence is to rely on the sojourn time of vehicles in the cell. The sojourn time is a crucial element in the characterization of user mobility in cellular networks [25, 26]. However, in the literature, only a few studies investigate the sojourn time in the vehicular context [27]. In general, vehicular mobility and traffic patterns are inferred from mobile phone connections, indicating travel patterns [27]. Other techniques, based on vehicle detectors and GPS probes, measure the residence time of cells [28]. In this study, contrary to sojourn times that characterize vehicles' behavior on an individual basis, we look into the behavior of a set of vehicles.

We assess the persistence of the potential offloading capacity directly from a set of nodes using vehicular mobility datasets. First, we introduce a reduction threshold α to characterize the point we assume a set has dissolved. Then, we evaluate the persistence of nodes between their formation and dissolution – either after a short time (low persistence) or after a long time (high persistence). We also assess the impact of the size of cells by varying it. Finally, we point out the main differences between the datasets and their impact on the potential offloading capacity of the vehicular clouds. The primary outcome of our analysis is the characterization of periods of time that should be reserved for light tasks and periods has are keen to offload tasks that require longer computation time, thanks to a more stable potential offloading capacity.

Published and submitted works related to this contributions are:

- S. Khizar, M. Dias de Amorim, and V. Conan, "Persistence of Vehicular-Augmented Mobile Edges", *IEEE PIMRC*, London, United Kingdom, 2020.
- S. Khizar, M. Dias de Amorim, and V. Conan, "Un MEC qui persiste au-delà de la bordure", *CoRes*, Lyon, France, 2020.
- S. Khizar, M. Dias de Amorim, and V. Conan, "Vehicular Persistence: Estimating the Capacity of Vehicular Networks to Extend Multi-Access Edge Computing", submitted to a journal, 2021.

Contribution 2: Leveraging mobile devices in a pedestrian scenario

We specifically address the impact of device mobility on task execution success in a pedestrian scenario. We characterize a task as a *duration*, i.e., the time it takes for a mobile node to complete the task. We further consider that a mobile device must complete the task within some given time frame, which we call *completion delay*. To assess task execution success, we compare task duration with the node presence time in the cell. It allows us to obtain the *potential* number of nodes that would be able to execute the task. Our analyses provide an assessment of node mobility's impact on the task offloading potential of a cell. On the one hand, the completion delay impact shows that the more significant the completion delay, the more possibilities to execute a task. On the other hand, continuously increasing the completion delay is unnecessary because the gains become negligible after a certain point. Thus, depending on how fast the MEC needs to execute tasks, it can identify the maximum number of tasks that mobile nodes can perform. Our other results indicate that even the nodes that repeatedly enter and exit a cell can support the MEC task offloading.

Published and submitted works related to this contributions are:

- S. Khizar, M. Dias de Amorim, and V. Conan, "Offloading computing tasks beyond the edge: A data-driven analysis", *13th IFIP Wireless and Mobile Networking Conference*, Montreal, Canada, 2021).
- S. Khizar, M. Dias de Amorim, and V. Conan, "Quand les noeuds mobiles passent à l'action !", *CoRes*, La Rochelle, France, 2021.
- S. Khizar, M. Dias de Amorim, and V. Conan, "Resource quantification for task offloading beyond the MEC: A data-driven analysis", submitted to a journal, 2021.

In sum, both questions are closely related; temporal and spatial aspects go hand in hand in this study. The significant difference between the two works lies in the type of mobility and the angle of approach. Furthermore, one focuses on analyzing the evolution of a group of mobile devices and the other on analyzing the individual mobile device. Finally, thresholds are used in both studies to understand the impact of mobility better. One ensures that there are enough mobile devices present, and the other tries to recover more resources. Our way of considering this topic is summarized in Figure 1.4.

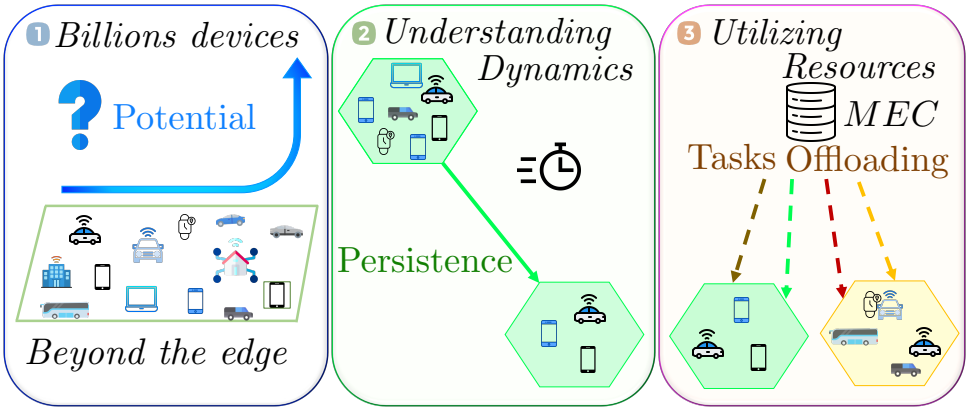


Figure 1.4: Our journey to reflection on this topic.

1.6 Thesis outline

The remainder of this thesis is organized as follows. In, Chapter 2, we describe the general background knowledge regarding the topic. This step is all the more crucial given the very varied literature in this field. In, Chapter 3, we start by focusing on the temporal availability of beyond-the-edge resources. Precisely, we investigate the co-location of mobile devices and analyze their persistence in a cell. In, Chapter 4, we extend our investigation on the spatial aspect by quantifying the resources that a cell can provide to perform a MEC task. Thus, we estimate the potential number of computational tasks performed by nodes based on the cumulative presence time in a given cell and completion delay. Finally, we conclude the manuscript and give directions for future work in Chapter 5.

—Change is the law of life. And those who look only to the past or present are certain to miss the future—

John F. Kennedy

2

Positioning and Related Works

This chapter outlines the main motivations and issues that underpin the importance of the MEC paradigm. We then discuss the key characteristics of this new computing paradigm. Finally, this chapter examines the main constraints essential to deploying MEC in an existing network infrastructure. This chapter also presents the current state and problems of offloading computation on the MEC. These concepts are essential for comprehending our positioning and the remainder of the work provided in this thesis.

The ongoing need for increasingly sophisticated mobile devices has driven the industry to produce technological advances and promote significant improvements in mobile device capabilities. In recent years, several reports published by Cisco or Ericsson have shown that these devices' hardware capabilities improved considerably [1, 29]. For example, a recently launched mobile device can easily outperform many desktop PCs in the past decade in performance and communication efficiency. Moreover, these mobile devices embed more advanced sensors. Nevertheless, the increase in battery life has not been as advanced as the other parts. For example, the lithium battery technology proposed ten years ago is still in use. However, overcoming battery issues while meeting the ever-increasing number of mobile devices and compute-intensive applications is challenging. As a result, several studies have devoted considerable effort to developing software-level

optimizations that support resource-intensive tasks on resource-constrained devices.

2.1 Mobile Cloud Computing

The concept of *computation offloading* stands out as a core mechanism enabling mobile users to migrate computationally and resource-intensive tasks to an external provider of computing resources rather than the mobile device itself. Furthermore, it enables solving the inherent problems of mobile computing by using scalable computing resources of the cloud servers without time and space constraints [30]. As a result, mobile devices can run compute-intensive applications using the cloud's computing and storage capabilities — this usage introduces *mobile cloud computing* (MCC).

Since its formal definition and standardization, cloud computing (CC) has evolved dramatically. It has been widely deployed in the past decade. Softwarization and virtualization technologies centralize resource management. They give flexibility and a supply of processing power, storage space, and network resources across distributed locations. Thus, CC resources are promising to support the emergence of diverse and sophisticated mobile applications enhanced by the revolution in cellular networks and mobile devices capacity. However, the MCC framework is distinct from the traditional CC paradigm. Indeed, MCC visualizes service provisioning through a collection of mobile-aware computational and information services. Awareness in a mobile computing environment includes various features such as energy consumption evaluation, location, interest awareness, wireless network cost calculation, and trace prediction. The MCC paradigm combines CC with Mobile Computing to provide cross-platform resource provisioning at a significantly lower cost. It brings many advantages for mobile devices besides its original purpose of overcoming battery constraints. We summarize the main benefits of MCC [31, 32, 33] as follows:

- Extending battery lifetime by offloading energy-consuming computations of the applications to the Cloud.
- Enabling sophisticated applications to the mobile users.
- Offering the mobility of location-independent computation.
- Providing higher data storage capabilities to the user.

However, the considerable distance between end devices and the MCC causes several issues:

- High latency: In MCC, data created in end devices must be sent to the Cloud to be processed. After that, MCC has to return the process to the end devices.
- Performance depends on network conditions: The more network hops between the end device and the Cloud, the worse the network state, which is terrible for the user experience.
- Network congestion: Billions of Internet-connected devices are on the market and continue to increase, so they contribute to Internet traffic, which will become a burden on the network backhaul.

In addition, several factors can reduce the benefit of the offloading method, especially bandwidth limitations between mobile devices and the distant Cloud and the amount of data transmitted between them. Researchers have already proposed many techniques for optimizing offloading strategies to improve computational performance and save energy. For example, users can decide to offload some or all of their computations to cloud servers for data processing instead of using only the mobile device [34, 35]. Barbera et al. proposed that time-sensitive tasks should perform in the cloudlet, and those with more flexible time requirements might execute in the Cloud [36]. Liu et al. demonstrated that computationally-intensive tasks are a better fit for cloud execution [37]. High-volume tasks with lower demands on computing resources are suitable for user equipment execution. Thus, despite such an improvement for mobile devices, MCC fails to fulfill the low-latency requirements of emerging mobile applications due to the bandwidth limitation between mobile devices and the distant Cloud and the amount of data transmitted between them. Moreover, the delay induced by the distant Cloud cannot tackle time-sensitive applications, such as online mobile gaming, augmented reality, face and speech recognition. Therefore, to solve the high latency issue, cloud services need to shift closer to the end-users. Recently, the emerging paradigm is computing at the edge of the mobile network.

2.2 Multi-access edge computing

As one of the most promising technical approaches to improve the effectiveness of cloud computing, multi-access edge computing (MEC) has attracted considerable attention recently. The MEC shifts the cloud services to the radio access network and provides the cloud-computing capability closer to the mobile users [2, 5, 38, 39, 40]. This proximity ensures low latency connection in the computation offloading process, thus improving the

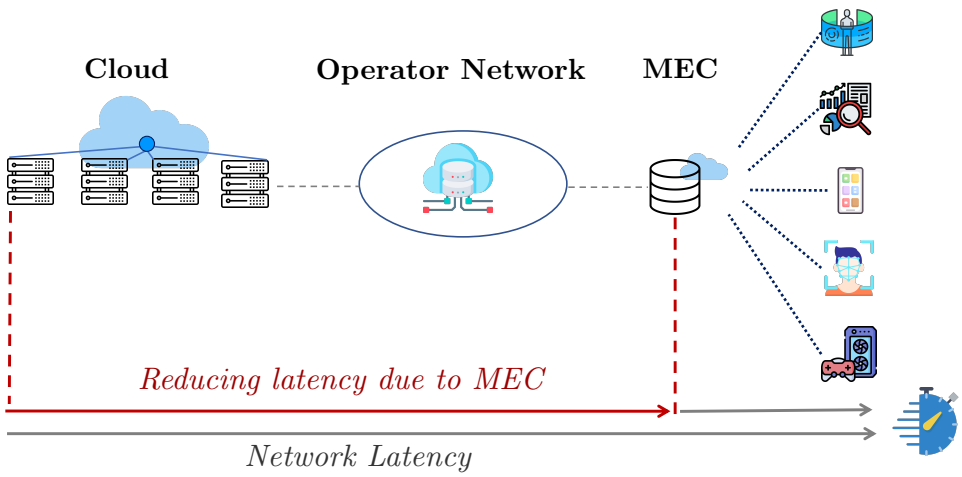


Figure 2.1: MEC architecture enhancing applications and low latency.

quality of services expected by users [2, 38]. Furthermore, MEC performs computational tasks and supports offloading data from mobile applications and services near the end-user (see Figure 2.1). As a result, considerable research in MEC computation offloading in cellular networks has been proposed in recent years. System performance gains, such as reducing the energy consumption or system cost by optimizing offloading decisions and allocating the resources effectively, are the main target of MEC technology. In addition, recently undertaken work enhances MEC with many features and functionalities such as connected vehicles, video streaming, video gaming, extensive data analysis, and healthcare [14, 41, 42].

Various architecture with similar concepts has been introduced to tackle challenges. Edge computing has been proposed to provide an intermediate layer between mobile devices and the cloud. This layer is implemented with few differences depending on the usage scenarios and locations. There are three leading architectures in the literature in edge computing: fog computing, Cloudlet, and multi-access edge computing.

The concept of Cloudlet was proposed by an academic team from Carnegie Mellon University, and the term “cloudlet” was first thrown by Satyanarayanan et al. [43]. It is considered an extension of the cloud as depicted in Figure 2.2. It is a trusted and resource-rich node with stable Internet connectivity. Cloudlet offers computing and access, and storage resources to nearby mobile devices. Furthermore, we can qualify the cloudlet concept as a small-box data center using a virtual machine-based approach. Cloudlet is usually deployed at one wireless hop away from mobile devices, such as public places like hospitals, shopping centers, and office buildings. Cloudlet is presented as a promising

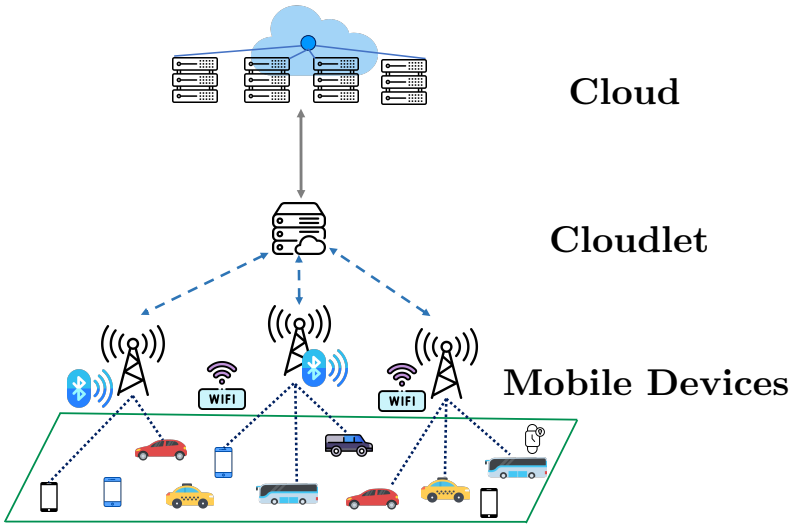


Figure 2.2: Cloudlet.

solution for remote wide-area network latency and cellular energy consumption through cellular data connections to the cloud. Nevertheless, Cloudlet relies on a solid Internet connection since it uses technologies such as Wi-Fi, which is placed one hop or many hops at the Internet's edge. Furthermore, particular security and privacy challenges entail consumer reluctance to utilize privacy-related services, such as e-commerce websites. Since it can be deployed independently in different locations, managing it is difficult.

On the other hand, the concept of Fog computing, also known as "fog networking" or "fogging", was introduced by Cisco to enable mainly the support of Internet of things applications [44, 45]. Fog computing uses edge devices such as switches and routers to perform offload functions on wireless networks rather than on separate servers. Moreover, the owner edge server in both of cloudlet and fog node can be anyone. In addition, fog computing has limitations because it relies on wireless connections that need to be live to perform complex actions. The terms fog computing and MEC are often used interchangeably, but they differ in some respects. We show the main difference between fog computing and MEC in Table 2.1.

Finally, the third approach, namely Multi-access edge computing, is a standardization developed by ETSI group. Unlike the first two implementations, the ETSI group defines algorithms and standards that describe how the MEC infrastructure is designed, deployed, and used. MEC architecture essential components are mobile devices such as smartphones, intelligent sensors, connected vehicles, MEC servers, and wireless base stations such as Wi-Fi access points, 3G/4G/5G). Multi-access edge computing provides cloud computing capabilities to radio access networks and hosts applications at the mobile

Characteristics	Fog	MEC
Node devices	Gateway, access Point, routers	Base station
Node location	Varying between mobile devices and Cloud	Radio Network Controller or Base station
Software architecture	Fog abstraction layer based	Orchestrator based
Proximity	Single or multiple hops	Single
Context awareness	Medium	High
Access Technologie	Wi-Fi, bluetooth, mobile networks	Mobile networks

Table 2.1: Difference between Fog computing and MEC.

network edge, improving user experience. The benefits of the MEC architecture compared to the traditional centralized cloud computing are as follows:

- Reduced latency: Communication latency reduces significantly because the end-users processing and storage functions are close. As a result, user experience is considered high quality with ultra-low latency and high bandwidth.

- Proximity: MEC benefits latency-sensitive applications and computing devices such as augmented reality and video analytics by distributing resources closer to the end-user.

- Network Context Awareness: By incorporating MEC into their business model, applications that provide network information and services based on real-time network data may benefit businesses and events. These applications can estimate radio cell utilization and network bandwidth throughout real-time RAN information.

- Energy efficiency: The hardware components of mobile terminals, such as the screen, CPU, and network interfaces, are more energy-intensive. Moreover, one of the most crucial issues with today's mobile devices is battery life. Because manufacturers have difficulties fulfilling rising demand, MEC is a potential alternative by bringing cloud computing capabilities closer to mobile terminals to extend battery life.

- On-demand self-services: MEC provides resources for mobile devices in just seconds. However, end-users may be required to make expected computing resources reservations for some complex applications and services.

MEC is a cost-effective and highly attractive extension of cloud computing. It can dramatically improve the performance of numerous applications and services and enhance the user experience. One of the main benefits of this paradigm is to translate multiple applications from prototype to development, and large-scale use, such as mobile gaming and mobile augmented reality apps. In addition, MEC provides computing resources everywhere, allowing end-users to run where it works best for them. However, real-world deployment and use of MEC requires tackling complex challenges. Indeed, integrating a MEC platform into a mobile network environment poses several challenges related to service orchestration, primarily due to fluctuating resources and changing radio and network conditions caused by user mobility. Furthermore, a MEC system requires support for managing the application lifecycle, i.e., instantiating and ending an application, either on-demand or responding to a request from an authorized third party. Orchestrating a MEC platform in resource allocation and service placement is essential to ensure efficient use of network resources, optimal quality of service, and reliability. The following service attributes are particularly relevant to the efficient operation of a MEC service orchestration process. Computation offloading is considered a crucial use case from the user's perspective. It enables the execution of sophisticated new applications from mobile devices while reducing their energy consumption.

2.3 Offloading to the MEC

Computing Offloading from mobile devices to MCC or MEC is a well-established concept for supporting mobile devices with limited resources. Computation offloading is a mechanism in which resource-constrained mobile devices fully or partially offload their computation-intensive tasks to a resource-rich cloud to be executed in the cloud environment and obtain the results back. As a result, it saves battery life on mobile devices and speeds up the computation process. Many works have investigated this issue in the literature, introducing several new approaches with different offloading policies and algorithms. There is three mains taxonomy of computation offloading :

- Architecture aspect: This aspect defines the architecture of the MEC-enabled application. The works on this aspect define methods and designs to know how to offload an application and which parts. For example, we can offload different parts of an application to the MEC, such as a task, thread, class. We distinguish two categories of applications supported by MEC, the single task application, and the multi-task application. The first one is just one task that is offloaded to the MEC.

The second one consists of many tasks that can be performed on the MEC or some locally.

- Resource allocation aspect: A MEC application requires two sorts of resources: one for data volume and the other for its computation processing. However, mobile devices must consider CPU and bandwidth resources when offloading data and not only the computation resources aspect. Therefore, it has to reserve an amount of the network resource to transmit data to the MEC. Thus, both radio transmission and computing resources are essential for task offloading on the MEC.

- Performance aspect: This aspect is relevant to the offloading performance measurement. Metrics are critical indicators of offloading decisions. A key metric is application completion time, including quantifying time when an application starts, transmission, and processing time until it finishes. Minimizing this metric is essential for the effectiveness of computational offloading. Moreover, there is also the measure of energy consumption whose relevant to analyze the performance aspect. It measures how much energy a mobile device consumes while the application is running of the application. The goal is to ensure low energy consumption to prolong battery life.

In sum, the three main points reflect the primary purpose of exploring offloading computing solutions. The goal is to minimize the mobile device's energy consumption and execution time while adhering to the maximum latency limit. Alternatively, finding the optimal compromise between power consumption and execution time.

Recent studies focus on minimizing computational offload latency. For example, Ren et al. discusse the computation offloading possibility of multi-user video compression applications to minimize the latency [46]. Furthermore, they express also taking into account partial offload scenarios. Using a one-dimensional search algorithm by Liu et al. allows achieving the goal of minimizing execution delay [47]. The proposed algorithm finds the optimal offload decision-making policy according to mobile devices' current state, such as buffer state, available processing resources, and radio link conditions between the mobile device and the MEC host.

Various approaches have been proposed to minimize the total energy consumption of mobile devices while satisfying a tolerable latency constraint. Kwak et al. [48] focused on an energy minimization problem for local computation and task offloading in a single-user MEC system. Chen et al. [49] proposed a peer offloading framework for autonomous decision-making. A pre-computed offline approach and an online learning approach are

used to solve the optimization problem. Further extensions from single-user to multi-user scenarios are also investigate [50, 51]. Regarding task offloading, Tong et al. [52] designed a hierarchical peripheral computing architecture according to the distance between the peripheral server and users. They proposed an optimal offloading scheme to minimize the heuristic task duration algorithm.

Several papers [20, 53] investigate the power consumption of mobile devices based on dynamic voltage and frequency scaling (DVFS). Energy harvesting strategies are also used to reduce energy consumption in computation offloading during the execution at the mobile device and data transmission over the radio network [54]. Chen et al. [15] considered the task offloading and resource allocation in mobile edge computing. For stabilizing all task queues at the mobile device and MEC server, dynamic task offloading and resource allocation policies were proposed using Lyapunov's stochastic optimization in [48, 55].

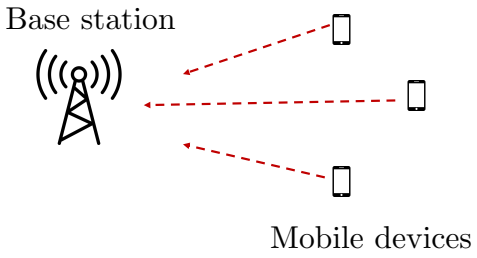
2.4 Discussion

Computation offloading is the central topic of many prior studies in mobile cloud computing (MCC) and mobile edge computing (MEC), as we already mentioned. Nevertheless, literature in the field lists several proposals focusing on tasks offloading in different architecture depicted in Figure 2.3. Thus, offloading tasks can be done:

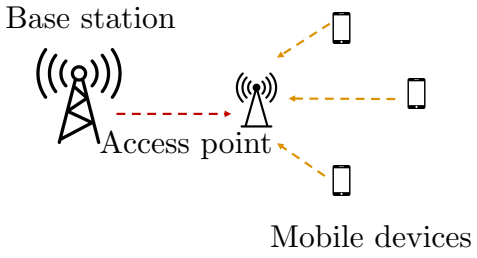
- To the infrastructure such as MEC servers [14, 15, 16, 17, 18, 56, 57, 58].
- Through a wireless Access Point [19, 20].
- Through mobile devices to mobile devices (i.e., between a coordinator mobile device and other mobile devices [21, 22, 56] or offloading between mobile devices (D2D) [23, 24].)

In our case, we envisioned a framework where mobile device proposes their untapped resources for multi-access edge computing task execution (Figure 2.3d). Mobile devices have been considered the weakest in the network infrastructure for a long time due to their low battery, low memory, and low CPU resources. Thus, they are supported by various cloud computing, fog computing, MEC.

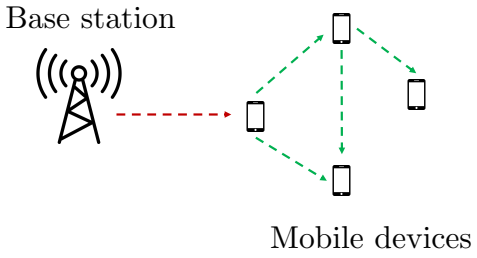
However, mobile devices have undergone a remarkable evolution from small devices with limited resources to essential accessories in everyday life. Over the years, they have gained more performance and improved computing, storage, and sensing capabilities. Thus, not a single day passes without using one or more mobile devices, and these several



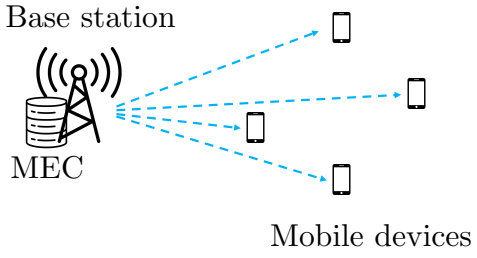
(a) Mobile devices to infrastructure.



(b) Access Point based.



(c) Mobile devices to mobile devices.



(d) **MEC to mobile devices**.

Figure 2.3: From conventional architectures towards leveraging mobile devices resources.

times a day. Mobile devices enables many activities, which explains their renown. They are helpful for both work (such as better organization, remote works facilities, communications) and personal uses (like sports and health monitoring, games, watching serials). As a result, mobile devices are ubiquitous and are changing our lifestyles and everyday activities. As a result of the increasingly global spread of advanced devices and existing work, we changed our perspective by considering that mobile devices can be a real potential of computing resources if they join their forces. Here, the main reasons that lead us to leverage mobile devices are:

- Continuous expansion: The number of mobile devices on the market is growing every year. Indeed, according to Cisco's forecast, there will be 29.3 billion networked devices by 2023 [1].

- Availability: Mobile devices are not always used to their total capacity and remain inactive most of the time [8, 9]. Thus, we can leverage them for MEC task offloading.

- Saving deployment cost: MEC needs to be deployed on a large scale to draw all its benefits [7]. It causes high costs. On the contrary, mobile devices are already available on a considerable scale, as mentioned above. Thus, it will cost far less to the operators to use them.

Moreover, we recall that the motto *"unity makes strength"* pushes us to believe that we may have insufficient capability by using one device, but a billion alters the picture.

2.5 Conclusion

In this chapter, we have provided the necessary background for understanding the benefits, the challenges, and the issues of computation offloading in MEC. One main advantage of computation offloading in MEC is to improve and enrich the performance of low-resource devices. Several papers have investigated these issues and presented various offloading policies and algorithms. This work improves the performance of the MEC application. However, most of this work considered mobile devices as the weakest link of the network infrastructure. Nevertheless, the growth of mobile devices and their diversification push us to take a look at them. In this thesis, we will focus on computation offloading in MEC and beyond the edge environment. Our works propose offloading tasks from the MEC to mobile devices. We investigate a new computation offloading strategy that identifies some factors that affect task offloading.

In the next chapter, we will start by exploring the importance of the co-location of mobile devices and their persistence in a given location. This metric is essential, as we need a deep understanding of the environment to determine whether it is possible to offload tasks.

—Success is not the absence of failure; it's the persistence through failure—

Aisha, Tyler

3

Persistence of Vehicular-Augmented Mobile Edges

As users are mobile, exploiting the idle capacity of their devices is challenging. In this chapter, we propose and investigate persistence as a metric to estimate the stability of sets of nodes in a cell and their capacity to offload tasks from the MEC. First, we investigate how nodes form sets according to their arrival time and co-location within a cell. We then assess the persistence of vehicular nodes using real-world datasets in Rome and Rio de Janeiro.

Operators seek to reduce latency while providing a high quality of service for end-users. Several technologies have emerged to tackle these requirements, including cloud and edge computing [38, 39, 15, 59, 60, 61]. As we already discuss, Multi Access Edge Computing (MEC) has recently been proposed to bring computing capacities closer to mobile users by shifting cloud services to the radio access network [2, 3, 4, 5, 38]. Considering that most end-user devices are idle for the majority of time [8, 9], a question that arises is *up to what extent could end-user devices help offload tasks from the cloud and enhance the overall capacity of mobile edge computing?*

We use mobile devices' untapped resources to perform tasks on MEC's behalf. A

specific case we address in this chapter is *vehicular edge computing* (VEC) [14]. VEC extends MEC with traditional vehicular networks. One of the challenges of such an approach is that, because of the mobility of nodes, we need to estimate the available resources (i.e., vehicles) in a cell when MEC is about to offload a task.

We focus on the computation offloading capacity that a set of nodes can offer. When a task is about to be offloaded, we use the number of nodes to estimate the "potential" offloading capacity of the cell. This potential offloading capacity varies over time and depends on the dynamics of the nodes. The core of our work is the investigation of the persistence of a set of nodes

In short, given the set of nodes in a cell at a given time, the *persistence* is the expected lifespan of this set before its size decreases to a given percentage of its initial size. We make several analyses around the notion of persistence. First of all, we assess the impact of the size of cells. It allows having an idea of the coverage area needed to offload a task of a specific size. Furthermore, it may allow better placement of the MEC servers. We investigate then that a set of nodes succeed to offload tasks of different sizes.

We assess the potential offloading capacity directly from nodes using vehicular mobility datasets from Rome and Rio de Janeiro. Finally, we point out the main differences between the datasets and their impact on the offloading capacity. Our analysis's main outcome is the characterization of time-periods reserved for short and long tasks. Through more stable potential offloading capacity, we identify periods that are more appropriate for offloading tasks requiring longer computation times, and conversely, for short ones.

As a summary, the contributions of this chapter are:

- **Persistence as a metric to estimate the capacity of vehicular networks to offload tasks from MEC.** We argue that persistence is a crucial element for understanding the possibilities of using end-user devices to extend a MEC's capacity. Therefore, we propose calculating the persistence time of a cluster of resources based on their cells and formation time.

- **Persistence in vehicular scenarios.** We assess the persistence using vehicular datasets in two cities: (i) 320 taxi cabs in Rome, Italy [11] and (ii) 12,000 buses in Rio de Janeiro, Brazil [12].

- **Analysis of the impact of cell size variation on persistence.** We vary the cell size from 1 km^2 to 64 km^2 and visualize the persistence of all cells using heatmaps, to help network planners determine the best placement of MEC servers.

- **Analysis of the ability to accomplish a task.** We evaluate the practical feasi-

bility of offloading tasks on such datasets by devising different scenarios. We send tasks requiring 1 to 35 minutes of execution time to all sets of nodes for all scenarios. For the first scenario, if at least one node in the set remains for the execution time required for the task, we consider the task well-executed. In the second scenario, a task is well-performed if none of the nodes leave during the task execution time. For each scenario, we obtain the percentage of achieving tasks by the sets of nodes, allowing us to know if datasets are in a position to receive them.

The remainder of this chapter is structured as follows. In Section 3.1, we introduce persistence as a metric for evaluating the offloading potential of a cell. We also present datasets and the processing methodology. Next, we introduce the persistence of sets in dense and sparse areas in Section 3.2 and assess the impact of the reduction threshold in Section 3.3. Then, we continue our study by varying the cell size in Section 3.4. Indeed, we provide the spatial distribution of persistence for different periods of the day. We discuss the impact of cell size variation on our persistence metric. Finally, in Section 3.5, we investigate the task processing capacities by the different datasets. We obtain a percentage of tasks carry out successfully the task in two given scenarios. Furthermore, we analyze the completeness of tasks according to different scenarios. We conclude the chapter in Section 3.6.

3.1 Persistence of sets of nodes

A set of nodes in a cell represents an *offloading potential*, i.e., a measure to help estimate the computing load that the set can handle. The problem is that, as nodes are mobile, the offloading capacity changes over time. It is thus necessary to understand the dynamics of the system so that the MEC orchestrator can have a rough idea of where and when it can offload tasks to nodes in a set. subsectionPersistence

We refer to the time T when the MEC offloads a task to a set of nodes as the *offset*. We assume that nodes receiving tasks from the MEC must remain in the cell until the task completion; otherwise, the task is considered lost. We also assume that nodes joining the set after the offset do not handle the task (only future task requests). Therefore, we focus the analysis on the computation offloading capacity and the *persistence* of the set, i.e., the time elapsed between the offset and the dissolution of the set.

We model a geographical area as a set of cells C. We note $s_{n,c,T}$ the set of nodes located in cell $c \in C$ at time T (offset) and $n \in$ the set identifier. Our goal is to investigate the persistence of $s_{n,c,T}$. Loosely speaking, it is the period between T and the time the size of

the set decreases below some threshold. This threshold is specified by a variable $\alpha \in [0, 1]$, which we call the "reduction threshold." The closer the α is to one, the less tolerant the offloading system is to the departure of nodes. Therefore, the MEC can choose the right time when the persistence fits best with the tasks it has to assign to nodes.

To compute the persistence, we first formalize the reduction rate $R(t, T)$ of a set at a time $t > T$ as:

$$R(t, T) = \frac{|s_{n,c,t}|}{|s_{n,c,T}|}. \tag{3.1}$$

We define then a step function that characterizes the time at which the set of nodes decreases below the reduction threshold:

$$\mathbb{1}(s_{n,c,T}, t) = \begin{cases} 1 \; if \; R(t,T) \; \geq \; \alpha, \\ 0 \; if \; R(t,T) \; < \; \alpha. \end{cases} \tag{3.2}$$

Finally, we define the persistence as:

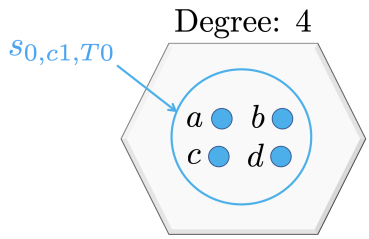
$$\mathcal{P}^{\alpha}(s_{n,c,T}) = \int_{T}^{+\infty} \mathbb{1}(s_{n,c,T}, t)dt. \tag{3.3}$$

In Figure 3.1, we illustrate the formation and dissolution of sets for α=0 (i.e., the persistence of a set between the offset and the time when *all* nodes in the set leave the cell). At $t = T_0$ (Figure 3.1a), four nodes are co-located in cell c_0, and they form the set s_{0,c_0,T_0}. The persistence of this set is three time intervals as the set vanishes at $t = T_2$. In Figure 3.1b, we show the configuration of the cell one time interval later. We notice that the cell contains, at $t = T_1$, five nodes, out of which two nodes come from the previous interval and three nodes that just joined the cell. This means that $s_{0,c_1,T_1} = 2$ and $s_{1,c_1,T_1} = 5$. The same principle applies to the remainder of the time.

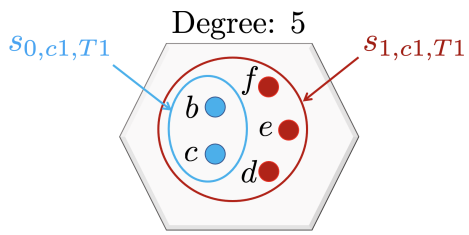
It is worth noting that the persistence of s_{1,c_1,T_1} is three time intervals because of $\alpha = 0$. For example, if $\alpha = 0.5$, the persistence would have been $\mathcal{P}^{0.5}(s_{1,c_1,T_1}) = 2$ time intervals because this is the time lapsed until the number of nodes decreases to half of the original set size.

3.1.1 Vehicular datasets

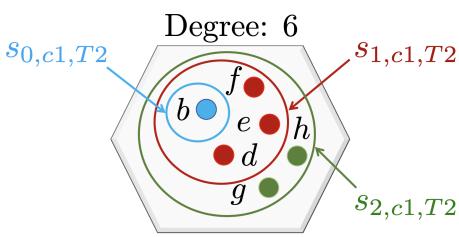
As technology has emerged dramatically during the last decade, vehicular networks have also been developed. They are becoming popular day by day due to cloud computing introduction and edge computing. Although there are few repercussions of this development, more computational capability and data communication are required. Over the



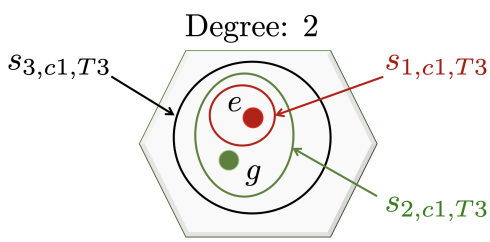
(a) At T0,$\mathcal{P}^0(s_{0,c_1,T_0}) = 4$



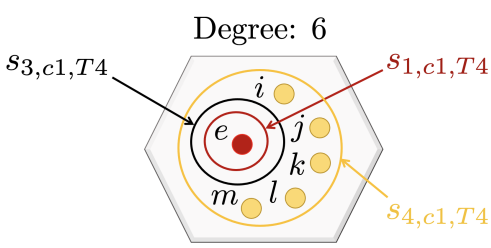
(b) At T1,$\mathcal{P}^0(s_{1,c_1,T_1}) = 5$, $\mathcal{P}^0(s_{0,c_1,T_1}) = 2$



(c) At T2



(d) At T3



(e) At T4

Figure 3.1: Formation of sets and their evolution over time.

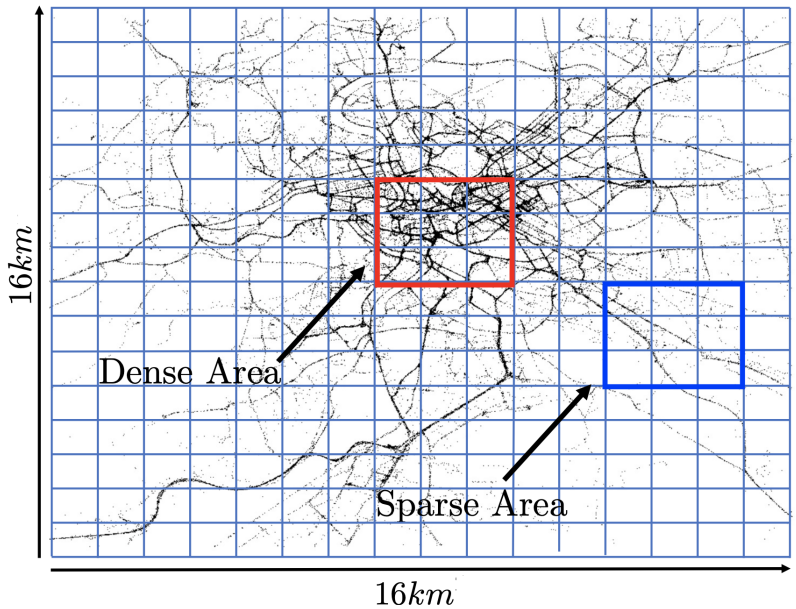


Figure 3.2: Simulation area of Rome (256 km^2).

last few years, the surge in vehicular network research has led to the proposal of countless network solutions specifically designed for vehicular environments.

Nowdays, *vehicular edge computing*–(VEC) is becoming an eminent trend [62, 14]. VEC integrates MEC with traditional vehicular networks. VEC aims to move part of the communication, computing, and caching resources to vehicles. It may relieve the congestion problem of backhaul links and offer untapped resources by running tasks on behalf of the MEC. Existing literature in the field lists several proposals focusing on computation offloading [14, 58, 57, 15] content caching [63], and security and privacy [64]. For example, several studies on vehicle networks explore the possibility of integrating WiFi access points into vehicles and, in some cases, cellular interfaces to allow vehicles to connect to surrounding vehicles. Sadek et al. proposed, for example, the use of a mobile vehicle gateway that uses WiFi for vehicle-to-vehicle communication (V2V) and LTE for vehicle-to-infrastructure communication (V2I). Another study introduced the concept of virtual access points. It extends roadside access points to vehicles that receive, store, and re-broadcast messages to uncovered areas. In short, there is a consensus that the use of vehicles is very relevant for various cases [3, 59, 39, 60]. The pertinence of mobility modeling for vehicle network simulation has been recognized for a long time. It drives the research community to seek more significant road traffic traces provided to network simulators. For our study, the two GPS traces that we use are vehicular mobility datasets, namely in Rome, Italy [11] and Rio de Janeiro, Brazil [12]. GPS technology

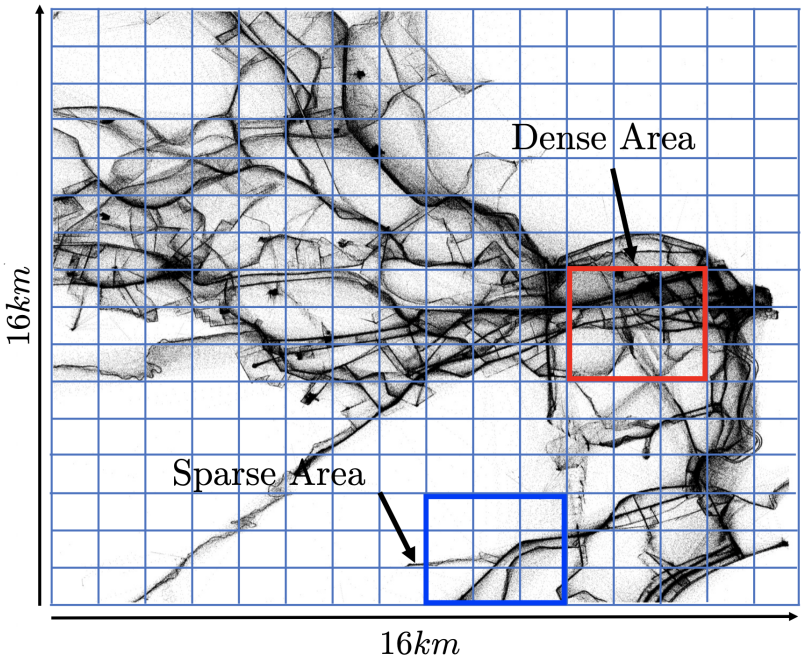


Figure 3.3: Simulation area of Rio de Janeiro (256 km^2).

enables tracking vehicles' locations and movements with the highest level of accuracy and temporal frequency.

The study of the persistence of a set of nodes during a 24-hour window using a granularity of one minute. We have selected weekdays. It is February 03, 2014, for Rome and October 01, 2014, for Rio. We have 272 taxis for Rome and 7,980 buses for Rio for those days. We obtain the position of the nodes through their GPS coordinates and evaluate the persistence in all cells of the two cities. For the sake of clarity, we present results separately for dense and sparse areas. Figs. 3.2 and 3.3 indicate these areas in Rome and Rio, respectively. The geographical area we consider is simulation areas of 256 km^2 for both cities. We subdivide the target areas into cells of varying sizes. In Figs. 3.2 and 3.3, we show 256 cells of 1 km^2 each. The dense and sparse areas cover 3 km^2.

3.1.2 Graphical view of persistence

Figs. 3.4 and 3.5, show the persistence of dense cells in both cities over a period of one hour. Note that each colored curve represents the dissolution of a set over time. Recall that a set starts at each time step and that its initial size corresponds to the total number of vehicles in the cell (black dashed line in the plot).

Let us pick the specific set formed at 7:03 am and describe its dynamics. In Rome, this set initially contains 11 vehicles at T =5:35 am. Consider α=0, which leads to

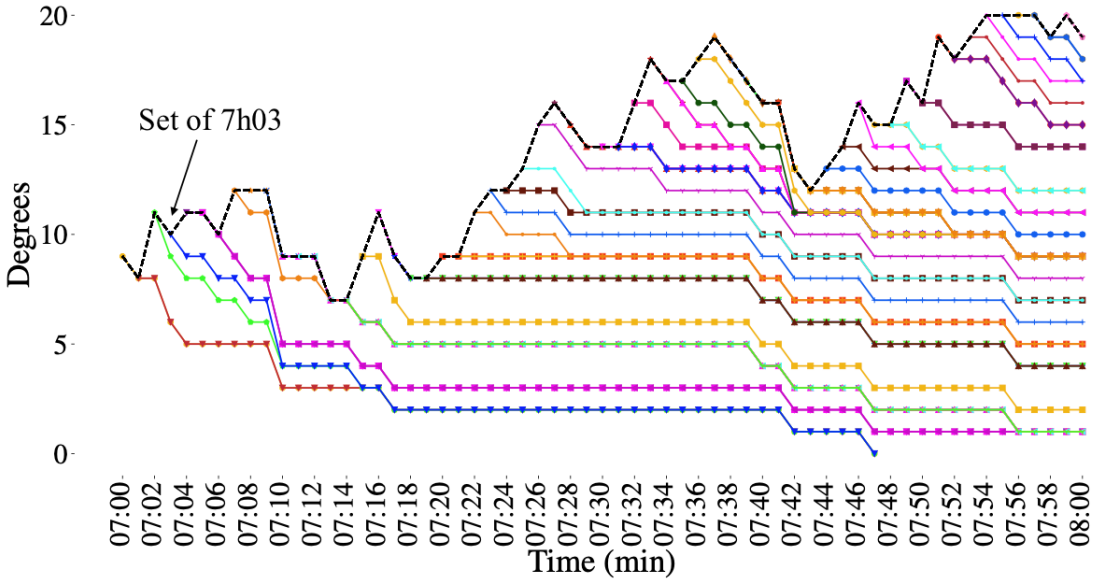


Figure 3.4: Persistence of a cell between 7:00 am and 8:00 am in a dense area of Rome.

persistence ranging from T until all nodes of set leave the cell; in this case, the persistence is 47 minutes. Similarly, let us consider $\alpha = 0.5$, the persistence is 10 minutes, which corresponds to the period between T until half of the set's nodes leave the cell. For example, in Rio, let us select the set formed at T =7:27 am; it initially contains 24 vehicles. The persistence with $\alpha = 0$ is 18 minutes while it is 6 minutes when $\alpha = 0.5$. Notably, Rio's persistence is lower than Rome's despite the larger number of nodes. There is a faster decrease in the curves leading to a shorter persistence.

Main outcomes: *The initial degree of sets of nodes is not necessarily related to the persistence time.*

3.2 Detailled analyses

We analyze the persistence in the dense and sparse areas of the datasets. To cover the full range of dynamics (from the creation to the dissolution of sets), we focus first on α =0 for the plots and show results for other values of α in the form of tables. We show in the plots the results on an hourly basis for a period of 24 hours.

3.2.1 Persistence in Rome (Taxis)

Traffic density in a metropolitan area varies over the day. In the following, we plot the median values of the persistence and the respective quartiles. As shown in Figure 3.6,

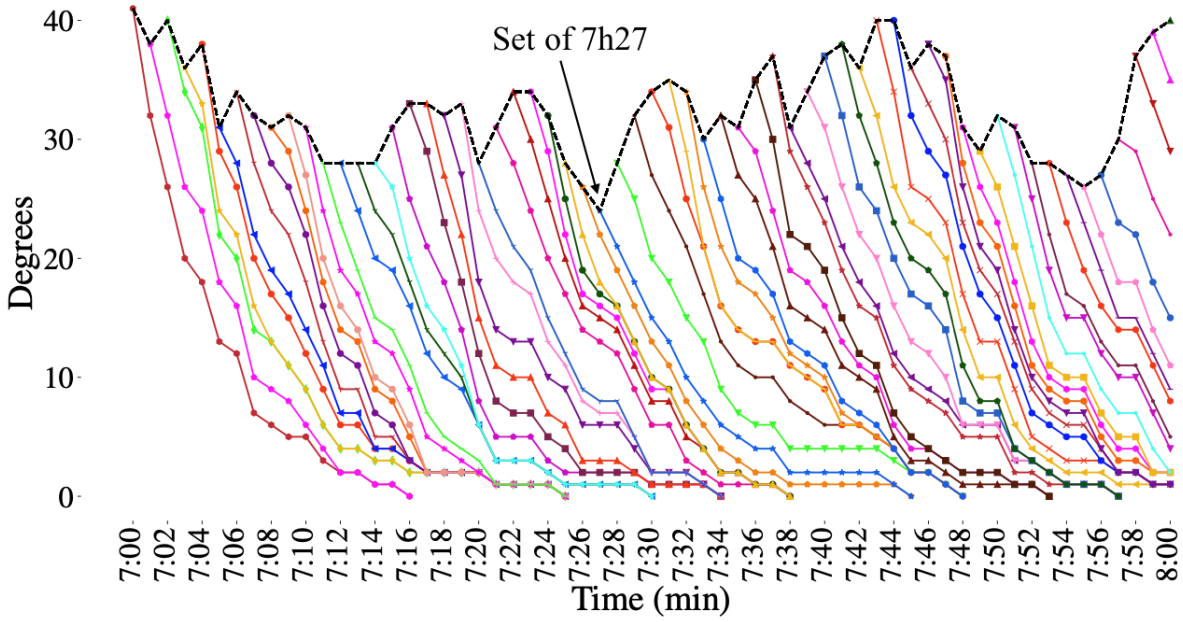


Figure 3.5: Persistence of a cell between 7:00 and 8:00 am in a dense area of Rio.

we observe in the dense area of Rome the presence of extreme values with relatively high persistence. The extreme values express the sets co-located during a significant period. The shortest persistence happens around 4:00 am. This interval is propitious to offload short computational tasks requiring low persistence (around ten minutes).

There is a persistence peak between 5:00 am and 7:00 am. The majority of the sets fall in this interval, and the median persistence is around 50 minutes. A gradual decrease of persistence is notable until it stabilizes around 20 minutes between 11:00 am and 16:00 pm. It indicates that the MEC could offload a task requiring high persistence in the early morning. We also observe significant values of the median during the interval from 6:00 pm to 10:00 pm, which implies a long persistence. These intervals coincide with the peaks of activities (leaving to work, to school, etc.). We find, as expected, that road loads have a strong impact on persistence time. We also have a considerable persistence between 2:00 am and 3:00 am, about 40 minutes. We can imagine that due to the nature of taxis, they are probably parked waiting for passengers.

During all day, persistence is low for the sparse area (Figure 3.7). The median persistence is between two and seven minutes. Furthermore, we observe extreme values indicating that a minority of nodes persist up to twenty minutes, but still, most sets have shorter persistence. We encountered exceptions for specific intervals from 5:00 am to 7:00 am, around 5:00 pm, 20:00 pm, and 01:00 am where the persistence is high. The median persistence is about 10 to 15 minutes and up to one hour for the third quartile.

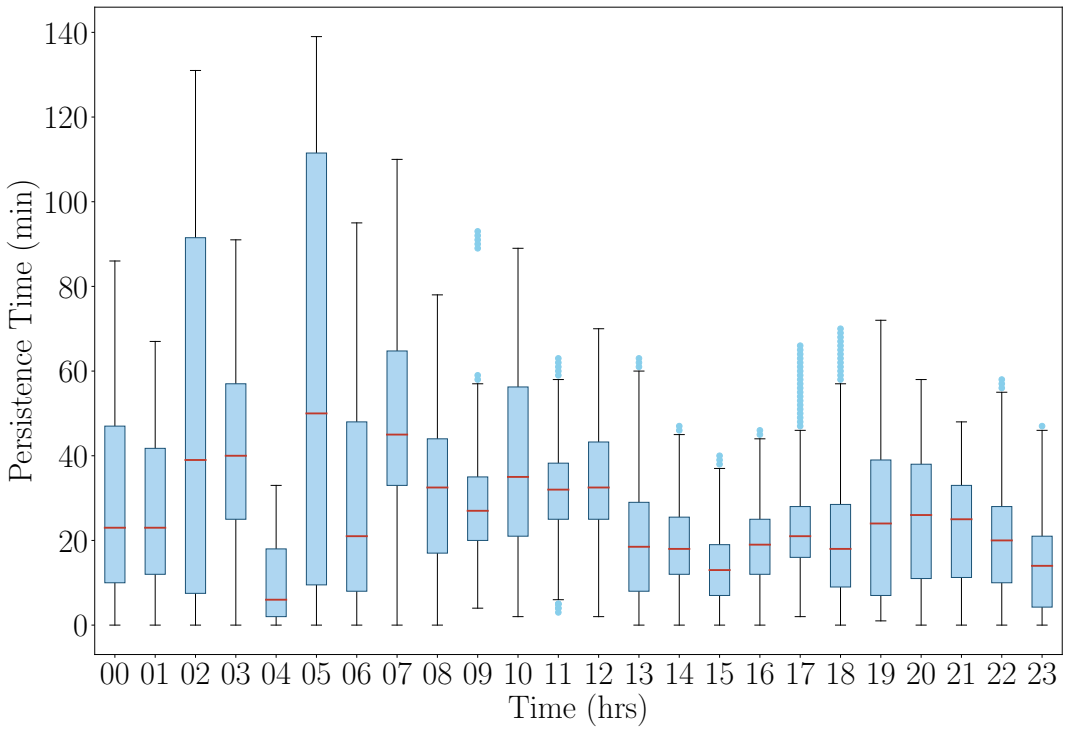


Figure 3.6: Downtown, dense area (Rome).

These specific intervals highlight the peak hours. We could argue that it is better to offload a computation task requiring low persistence in a sparse area except for the previous intervals.

3.2.2 Persistence in Rio (Buses)

Most buses start their service between 4:00 am and 10:00 pm. There is a high persistence in the dense area (Figure 3.8), from 12:00 am to 3:00 am, which is probably due to an end of service with bus parking, and those from 3:00 am to 4:00 am by a crossover of night services and day services. The strongest persistence happens around noon, with values going up to 320 minutes (more than five hours). Between 4:00 am and 6:00 am, we also have a peak of persistence; most sets are in this interval. Persistence is about 40 minutes, which facilitates the offloading tasks of this order of magnitude. We find high persistence in the morning from 4:00 am to 6:00 am then in the early afternoon from 1:00 am to 2:00 am and finally between 9:00 am to 10:00 am marking stream of people using buses at its hours. The median is between 20 and 40 minutes during the whole day, suggesting that the dense area of Rio is propitious for the MEC to offload tasks.

In the sparse area of the city (Figure 3.9), from 12:00 am to 4:00 am, persistence is

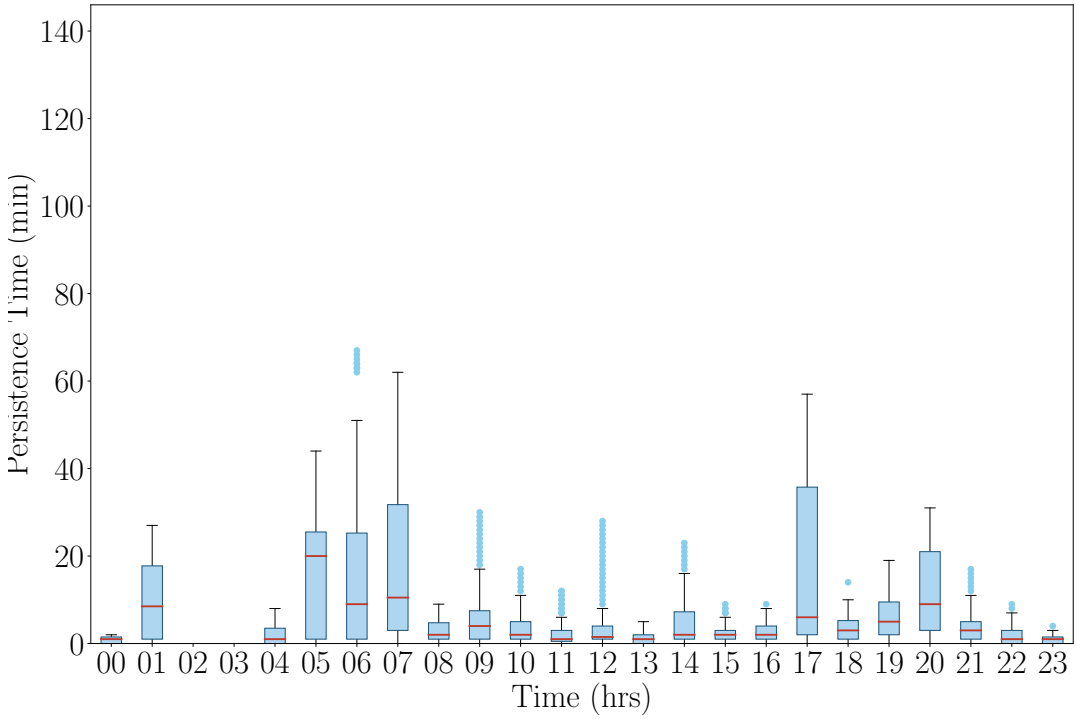


Figure 3.7: Periphery, sparse area (Rome).

close to zero and does not exceed five minutes. This interval may not be appropriate to offload a task requiring long computation time but rather for a short one. A median elevation is visible from 5:00 am to 7:00 am, the persistence remains relatively stable, around 40 minutes during the rest of the day. The highest median is around 4:00 pm with values up to two hours. After that, we observe a slight increase in the median from 8:00 pm to 10:00 pm, followed by a decrease. More buses are in service for times, such as 5:00 am, 12:00 pm, and 4:00 pm, and persistence is high also. Road traffic, especially on buses routes and shifts, directly impact the persistence of buses. In general, the median persistence remains low than the dense area. Nevertheless, 40 minutes of persistence seems enough to offload various task sizes while being in a sparse area.

Main outcomes: *Due to the nature of the vehicles, Rome taxis' persistence depends on the peaks of activity; in contrast, the persistence for Rio buses is less impacted by peaks of activities. Indeed, taxis operate at the customer's request, thus have no fixed itinerary. Therefore a taxi can go from one end of the city to the other several times a day. Their movements remain unplanned while buses follow a defined trajectory. Furthermore, the taxis are concentrated in the downtown area to have more customers, while buses, independently of the demand, make their trips in sparse and dense areas. Overall, we note that the dense zone has a higher persistence for Rome taxis and Rio buses. Thus the*

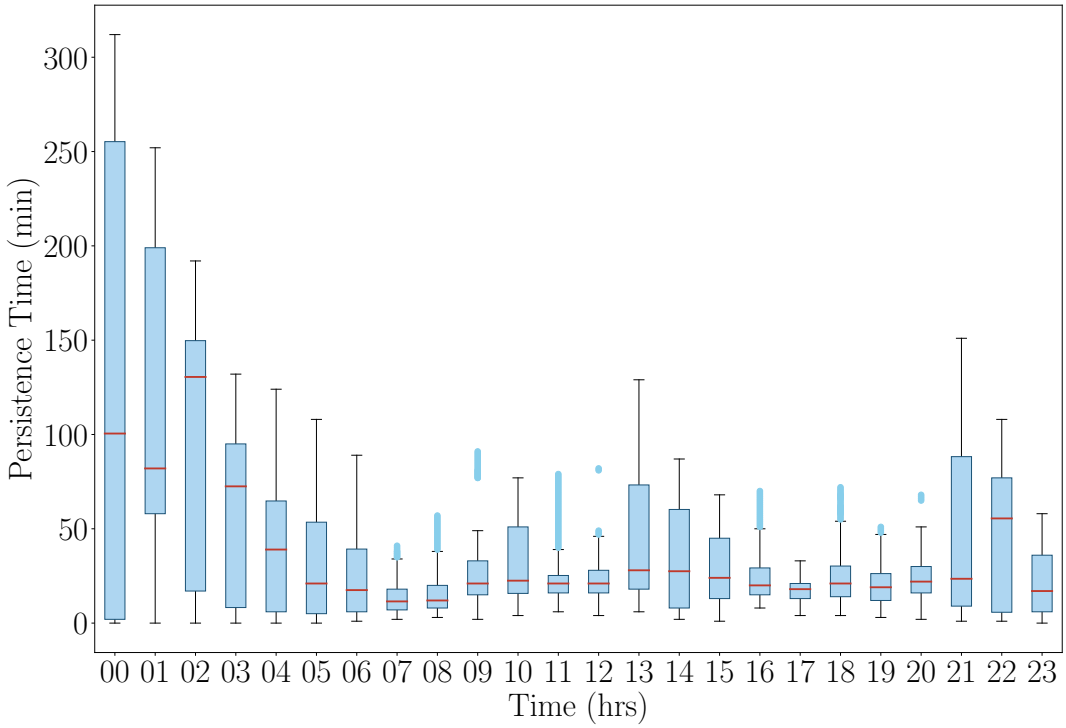


Figure 3.8: Downtown, dense area (Rio).

offloading of long tasks are therefore more applicable there.

3.3 Varying the reduction threshold

After studying the geographical distribution of persistence for $\alpha = 0$, we consider persistence with the following values of reduction threshold: $\alpha = \{0.25, 0.5, 0.75\}$. Firstly, we segment the day in 6 slots. For each time interval and dataset, we compare the dense and sparse areas. We present, in Tables 3.1 and 3.2, the maximum persistence compared to the median for the cities of Rome and Rio, respectively. Using the median and maximum persistence allows us to study the homogeneity or disparity, enabling a better reflection of the dispersion. We then observe the impact of the threshold on persistence.

3.3.1 Influence of the reduction threshold in Rome

We show the impact of the reduction threshold in the city of Rome in Table 3.1. For dense areas, the effect of the reduction threshold is visible for median and maximal persistence. For example, between 12:00 am and 6:00 am, the median persistence is equal to 35 minutes when α=0.25, 70 minutes when α=0.50, and 105 minutes when α=0.75.

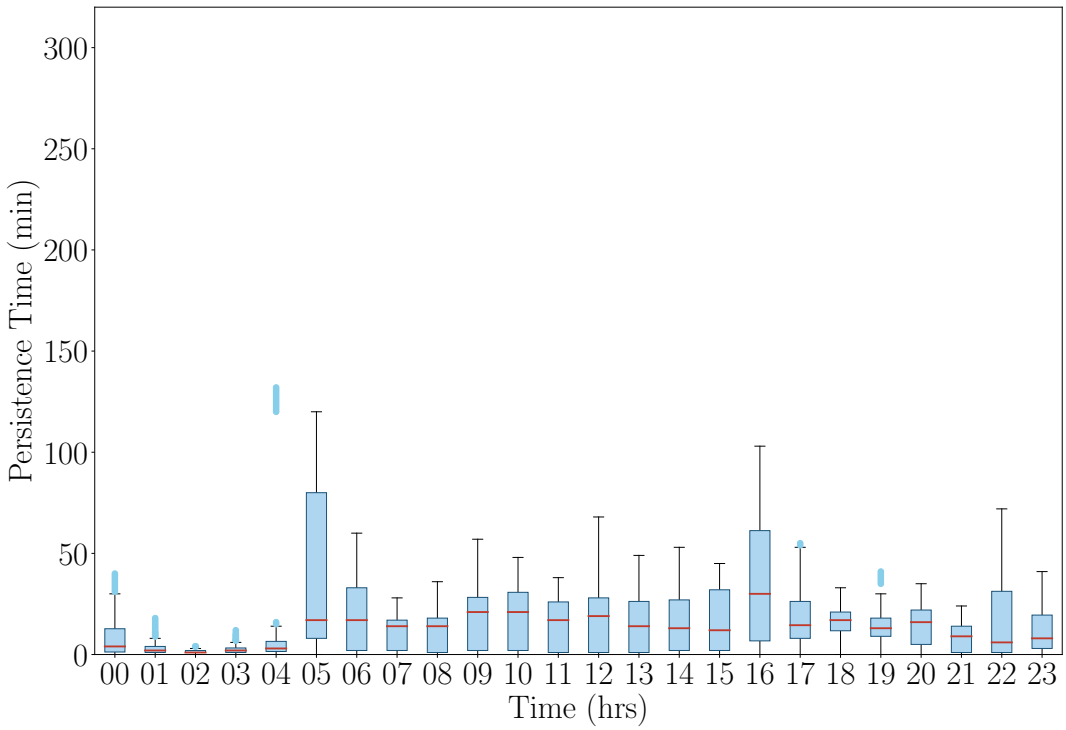


Figure 3.9: Periphery, sparse area (Rio).

We notice that for the reduction thresholds α=0.25, the maximum persistence values are identical for the 2:00 pm to 5:00 am and 5:00 pm to 8:00 pm intervals. However, the values are higher for the 5:00 pm and 8:00 pm intervals when we observe the median persistence. Thus, we deduce that the nodes persist more at this interval, and therefore it is preferable to offload a task at this period for the MEC. The highest persistence falls in the intervals from 00:00 am to 6:00 am, as seen in the previous section. Moreover, if we want 75% of the nodes in a group available to run a task, the node is available for 6 to 7 minutes. However, the extent between the maximum and median persistence is huge for all intervals, so the persistence time is not homogeneous for Rome's city in the dense area.

For the sparse area, the persistence is smaller at the periphery. Regardless of the maximum persistence and the intervals, we notice that most sets have around two-minute median persistence. Only one interval has a median persistence of up to 4 minutes when α=0.75 (from 5:00 pm to 8:00 pm). Moreover, beyond a given α, the threshold no longer impacts the persistence. Indeed, between 09:00 am and 2:00 pm, when α=0.25, the median persistence time is one and also for α=0.50. We deduce that few sets are persistent in the periphery, and the variation of α does not influence much the persistence time of the set. The persistence time is inhomogeneous for the city of Rome in the sparse area. We

Rome Persistence		Dense Area		Sparse Area	
Time Intervals		Max	Med	Max	Med
[12 am – 6 am]	**0.25**	35	7	17	1
	0.50	70	13	34	2
	0.75	105	19	51	3
[6 am – 9 am]	**0.25**	28	8	17	1
	0.50	55	14	34	2
	0.75	83	24	51	3
[9 am – 2 pm]	**0.25**	24	7	8	1
	0.50	47	15	31	1
	0.75	70	21	23	2
[2 pm – 5 pm]	**0.25**	17	5	15	1
	0.50	33	12	29	1
	0.75	50	14	43	2
[5 pm – 8 pm]	**0.25**	17	6	8	2
	0.50	36	16	29	3
	0.75	54	24	43	4
[8 pm – 12 pm]	**0.25**	15	6	8	1
	0.50	29	12	16	1
	0.75	44	18	24	3

Table 3.1: Reduction Threshold Variation for Rome

notice that the medium persistence in dense areas corresponds almost to the maximum persistence in sparse areas; thus, it is preferable to offload short tasks in the sparse area.

3.3.2 Influence of the reduction threshold in Rio

The impact of the reduction threshold is also visible for the city of Rio de Janeiro, as we can note when comparing Tables 3.1 and 3.2. For the interval 12:00 am to 4:00 am, from a specific α (α=0.25 and α=0.50), the reduction threshold no longer impacts the median persistence. Whereas from 7:00 am to noon, median persistence time is more uniform across all sets. Because of its uniformity, this interval is more suitable to receive calculation tasks for the whole set (around 10 minutes). The highest persistence time is between 12:00 am to 4:00 am. Depending on the alpha tolerance, we have a median persistence between 18 and 54 minutes, which appears suitable to offload long tasks.

Compared to the dense area, the persistence is low in the sparse one. Despite a huge maximal persistence, from 12:00 am to 4:00 am, the median persistence is only about 1 minutes. The dispersion of this interval is much larger compared to the median regardless of α. It means that there are many sets with low persistence; it is an off-peak time when the buses are certainly mostly inactive. Moreover, sometimes, α does not significantly impact due to the co-location time being too short or to the nodes staying

Rio Persistence		Dense Area		Sparse Area	
Time Intervals		Max	Med	Max	Med
[12 am – 4 am]	**0.25**	73	18	33	1
	0.50	156	36	66	1
	0.75	234	54	99	2
[4 am – 7 am]	**0.25**	31	6	33	3
	0.50	62	11	66	6
	0.75	93	17	99	9
[7 am – 9 am]	**0.25**	23	4	15	4
	0.50	46	7	29	8
	0.75	69	11	43	12
[9 am – 4 pm]	**0.25**	33	6	26	5
	0.50	65	11	52	9
	0.75	97	17	78	14
[4 pm – 7 pm]	**0.25**	18	5	26	5
	0.50	36	10	52	9
	0.75	54	15	78	14
[7 pm – 12 pm]	**0.25**	38	6	18	3
	0.50	79	11	36	6
	0.75	114	17	54	9

Table 3.2: Reduction Threshold Variation for Rio

too long. We observe that for the same maximum persistence values, for the 12:00 am to 4:00 am and 4:00 am to 7:00 am intervals. Nevertheless, the median persistence is higher for the second interval, showing us that the nodes remaining co-located for an extended period. The whole median in dense and sparse areas are around 10-15 minutes, which means we can offload a short task no matter the chosen interval and area.

Main outcomes: *The reduction threshold allows guaranteeing to the MEC that a certain number of nodes are present to perform a task. We observe that the more we tolerate nodes leaving a cell, the more we gain persistence time. Hence, having 50% of the nodes regarding the group formation time provides us around 10 to 15 minutes of median persistence for the dense area of the two cities. Nevertheless, sometimes the impact of the reduction threshold has little effect, especially in sparse areas where persistence is low.*

3.4 Impact of cell size on the persistence

This section analyzes the average persistence time for the whole 16 km^2 selected for this study for both cities. We obtain a much finer visualization using heatmaps, allowing us to precisely target the cells having persistence times suitable for the MEC orchestrator when it is about to offload a task. The pattern is roughly similar to the preceding section

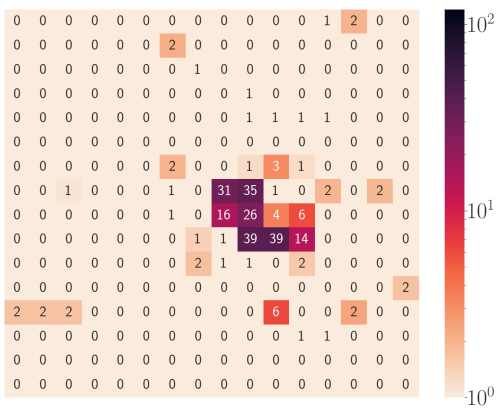
regarding persistence time for the dense and sparse areas. We then select a period and vary the cell size from 1 km^2, 4 km^2, 16 km^2, to 64 km^2. This variation allows us to understand the possible persistence gain by increasing the size of the cells so that the MEC orchestrator, according to its needs, will be able to make a trade-off between the persistence time required to execute a task and the coverage size needed.

3.4.1 Average persistence evolution in Rome using 1 km^2 cells

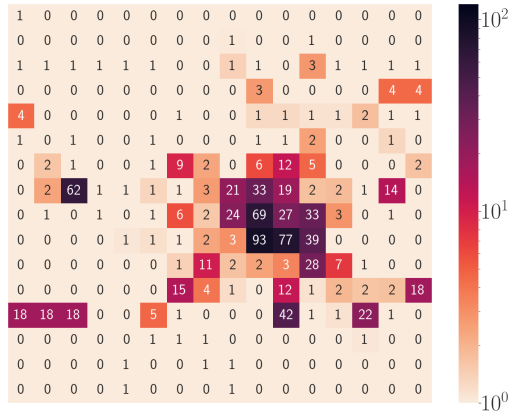
Figure 3.10 represents our simulation area for Rome. Each heatmap represents the evolution of taxi mobility over several intervals. Each square in the heatmap is a cell of 1 km^2. We have indicated the average persistence time in squares. Regarding the color, the darker it is, the higher the persistence is. The central area is the one with the highest persistence over time independently of the chosen interval. For instance, at $t0$, the persistence is almost zero on all the areas except the eight central squares representing the town center of Rome. These have persistence of about 30 minutes. As we progress in the day, the persistence in the periphery starts to increase. At $t1$, the persistence remains high in the downtown area, whereas at $t2$, we observe it is dispersed. Indeed, we find a more uniform distribution of persistence at this period in comparison to others. In addition to being balanced, it is also the period where persistence is important within the different cells surrounding the central area. We observe slight resorption of persistence at $t3$ and then an increase at $t4$. Finally, at $t5$, we have a general decrease in persistence and a return towards a concentration to the middle area. The viewing results support the fact that the co-location of vehicles is more important in the city center. The cab business is much higher in the downtown area due to the various clients found in crucial points such as train stations, monuments to visit for tourists or workplaces. Hence the opportunities to have high and reliable persistence are more present there. Transactions to the ring road are punctual and specific, so there is less chance for taxis to cross each other and have a high persistence, as we have seen. Another phenomenon we observe is that of the persistence extending more towards the northwest side. We believe that the movement is due to places such as the Vatican, airport and many hospitals present this side.

3.4.2 Average persistence evolution in Rio using 1 km^2 cells

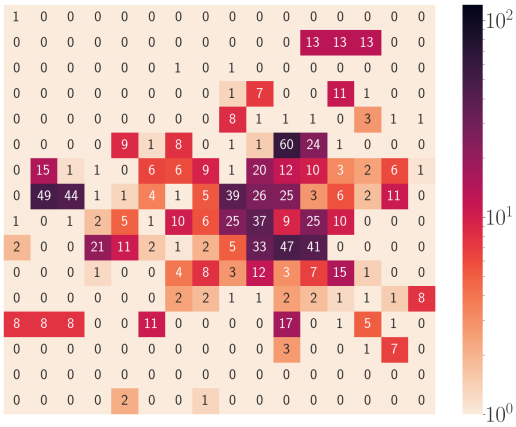
As in Rome, each heatmap represents the evolution of bus mobility over several intervals. It should be noted that the intervals for Rome and Rio are not identical due to the mobility of the two types of vehicles. Still, we find a pattern similar to that of Rome through these six intervals. Indeed, we find a low general persistence, then an increase



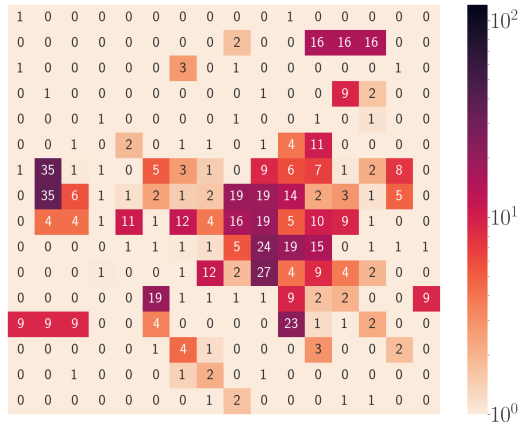
(a) At t0: 12:00 – 6:00 am



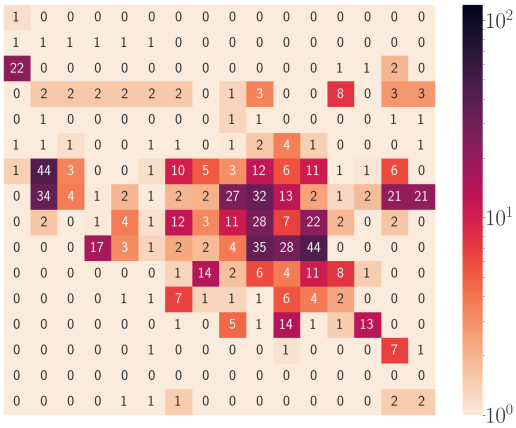
(b) At t1: 6:00 – 9:00 am



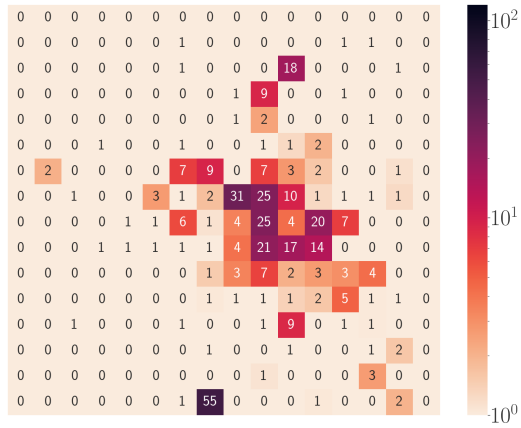
(c) At t2: 9:00 – 2:00 pm



(d) At t3: 2:00 – 5:00 pm

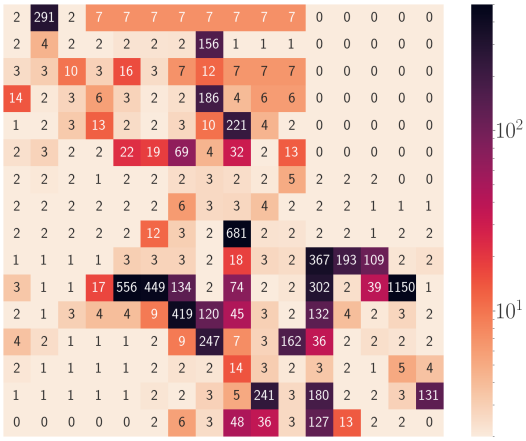


(e) At t4: 5:00 – 8:00 pm

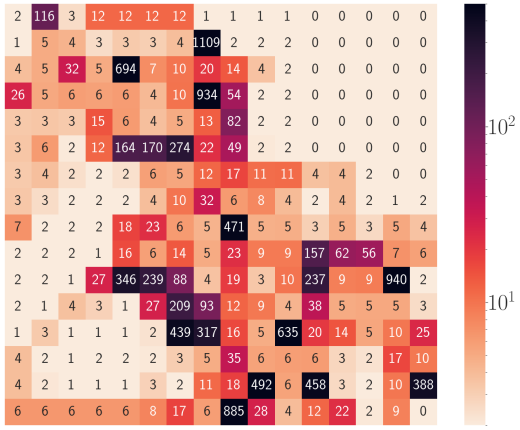


(f) At t5: 8:00 – 12:00 pm

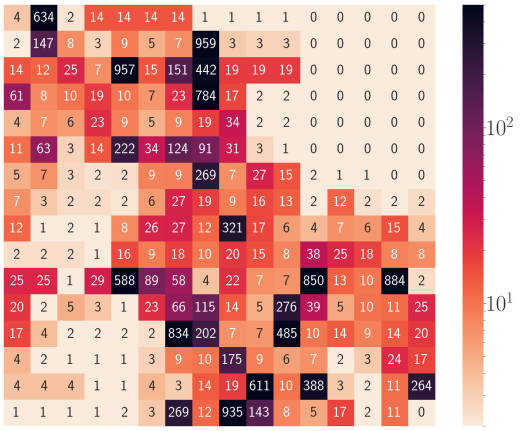
Figure 3.10: Average persistence of sets in Rome and their evolution over time for 1 km^2 cell size.



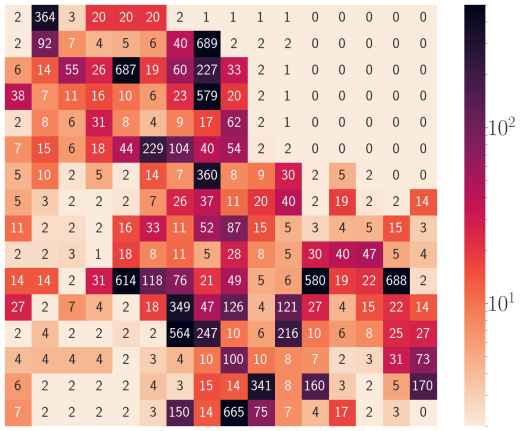
(a) At t0: 12:00 – 4:00 am



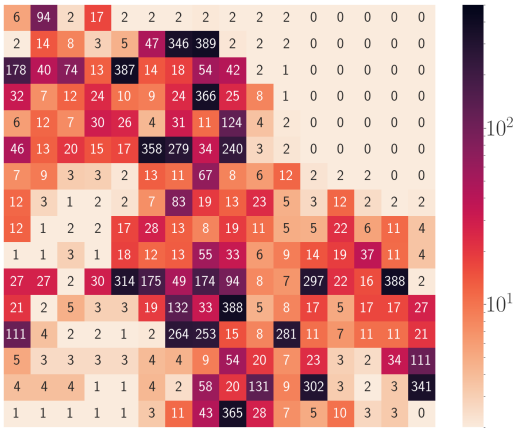
(b) At t1: 4:00 – 7:00 am



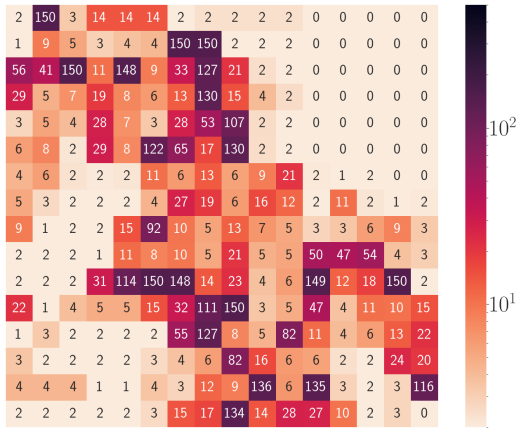
(c) At t2: 7:00 – 9:00 am



(d) At t3: 9:00 – 4:00 pm



(e) At t4: 4:00 – 7:00 pm



(f) At t5: 7:00 – 12:00 pm

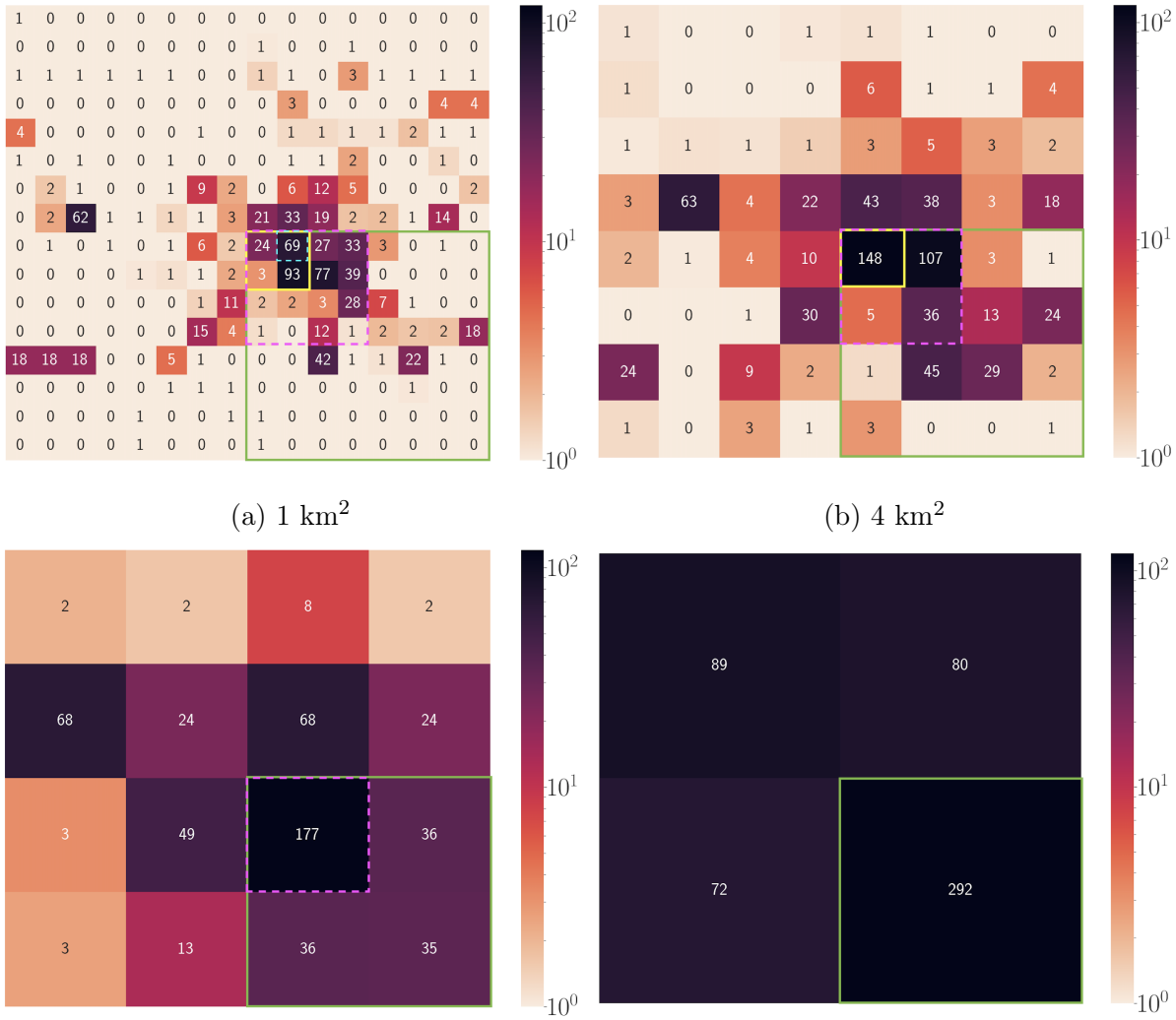
Figure 3.11: Average persistence of sets in Rio and their evolution over time for 1 km^2 cell size.

over time, and finally resorption. Nevertheless, a striking difference is that persistence is more widespread. We assume that it is primarily due to the difference in the size of the two datasets (320 cabs - 12,000 buses) and the type of vehicles. We notice a very localized persistence and a very high, at $t0$, up to 1100 minutes. We can assume that these are parked buses. Some of them are probably active, which explains the lower persistence in some cells. We note that as the day progresses, the northwest region has increasingly high persistence, especially at $t2$, $t3$, and $t4$. Conversely, the southeast region has lower persistence. The time periods $t3$ and $t4$ are those where the persistence is the highest to offload tasks. The Southwest area offers meager persistence times throughout the day ranging from 1 to 7 minutes. This area corresponds to the National Park of the Tijuca Forest. Roadways are restricted, hence the low persistence. We also have the northeast area, which is at zero; this area refers to the Atlantic Ocean. In general, the highest persistence occurs in the central area, followed by the city's north. Unlike Rome, we observe much more extensive and high persistence, thus offering many opportunities both in persistence time and in cell to offload tasks from the MEC.

3.4.3 Variation of cells size in Rome

We investigate the evolution of the persistence according to the size of the cells. Figure 3.12 represents the heatmap from the Figure 3.11b. In order to facilitate the reading, we have targeted a period, $t1$ (6:00 – 9:00 am), and a specific zone. We have shown on each of the plots the representation of 1 km^2, 4 km^2, 16 km^2, and 64 km^2. The selected cell in the central area covers 1 km^2 and has average persistence of 69 minutes, as shown in Figure 3.12a Figure 3.12b shows the aggregation of cells in 4 km^2. This increase causes the average persistence to improve from 69 minutes to 148 minutes. By increasing the size of the area by one kilometer, we can almost double the persistence time.

According to the MEC orchestrator requirements, if 69 minutes of average execution are enough, selecting a zone of 1 km^2 instead of 4 km^2 not to waste resources will be preferable. However, suppose the MEC orchestrator wants to execute long tasks to multiply the chances of executing, it is necessary to deploy access points to cover a larger area. In addition, using Figure 3.12b and Figure 3.12c results, we confirm that the more we increase the area, the more we have a huge persistence (117 minutes to 292 minutes). Overall it appears that we get a homogenization of the persistence in addition to its increase. Of course, increasing the coverage area leads to a loss of precision. The cell size in Figure 3.12a or even Figure 3.12b allows us to determine where the groups persist more. On the other hand, in Figure 3.12d, we cannot distinguish between dense and non-dense



(a) 1 km^2

(b) 4 km^2

(c) 16 km^2

(d) 64 km^2

Figure 3.12: Cell size variation in Rome from 6:00 am to 9:00 am.

areas. It is up to the system's user to do the trade-off between a gain of persistence time and loss of accuracy.

3.4.4 Variation of cells size in Rio

We repeat the same procedure for the city of Rio. Compared to the city of Rome, the results obtained for the city of Rio were quite surprising. We had already noted that the persistence is more extensive than Rome and less homogeneous. Indeed we can have two cells juxtaposed but with very different persistence. For example, we can have such cells with the persistence of 940 minutes near a cell with 2 minutes. The increase of an area and the aggregation of cells does not necessarily lead to better persistence. If we average

cells with lower persistence with very high persistence, it will be mitigated and heavily impacted. We notice this process in Figure 3.13.

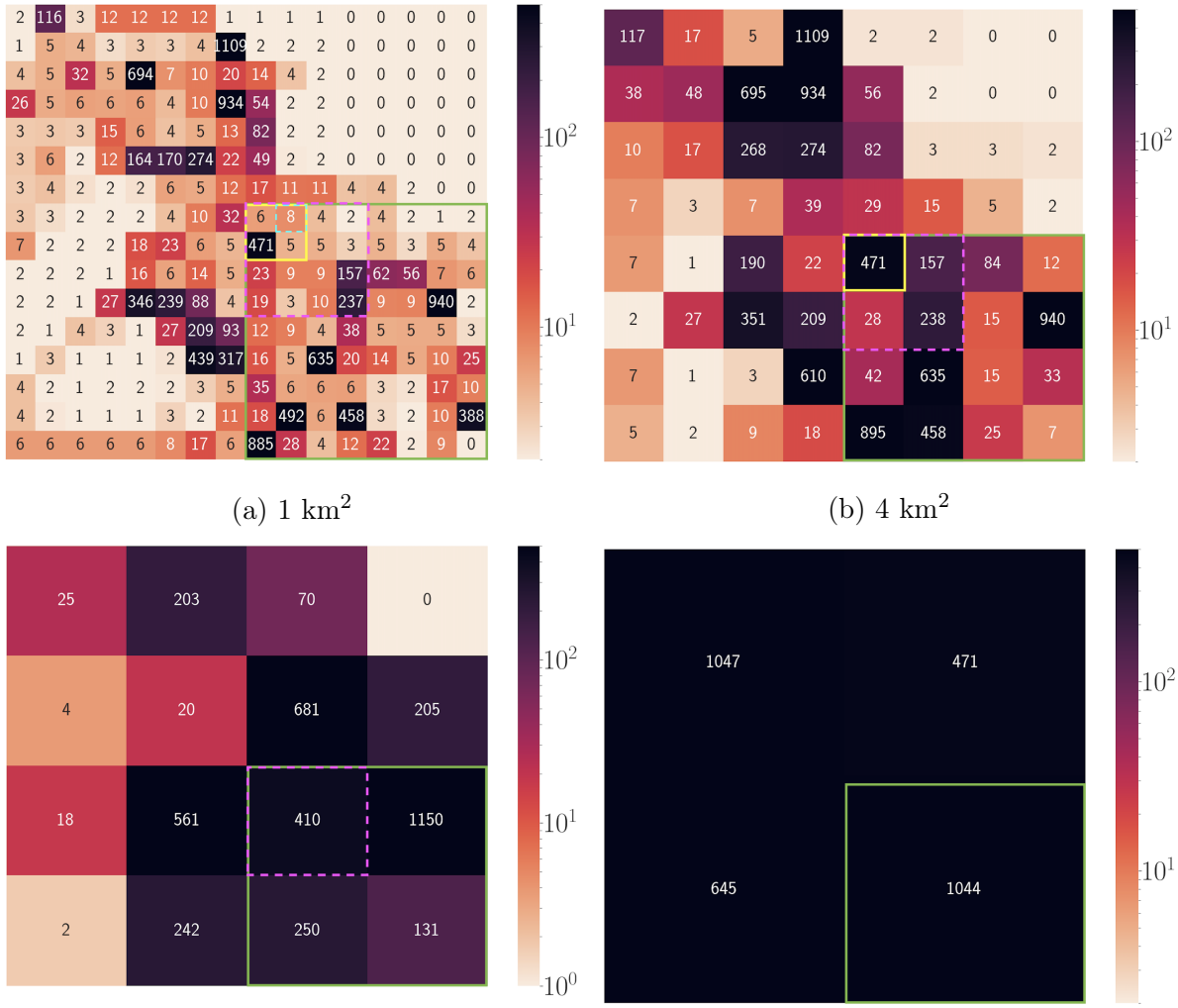
In Figure 3.13a we have four cells (just below the yellow frame) whose average persistence is equal to 23, 9, 19, and 3 minutes. Aggregating them gives a persistence of only 28 minutes. It is just five minutes more compared to the cell of 23 minutes. This case is the typical one where the trade between persistence time and precision is evident. Therefore, we would recommend sticking to a smaller cell size (1 km^2). To conclude, if the persistence is too heterogeneous inside a given area, the cell size increase may lead to a loss of precision and computation time. In that case, the MEC orchestrator should operate within smaller cells size to better target cells where a task can be handled. However, in general, we have an increase in persistence for a majority of cells when they are aggregated.

Main outcomes: *In general, the highest persistence occurs in the central area for both cities, as we already know. Nevertheless, in Rome, the persistence is very localized compared to Rio, where we observe much more extensive and high persistence. Moreover, we notice that sometimes the persistence gain is considerable when enlarging the size of the cells as in Rome, and sometimes it is minimal; this is particularly visible for the Rio dataset.*

3.5 Analysis of the ability to accomplish task

As a reminder, the persistence metric provides each set's lifetime. We refine our search to estimate the percentage of tasks performed successfully by the different nodes' sets based on their persistence. The obtained results are strongly related to the persistence of sets of nodes. It is evident that the longer a set of nodes persists, the more opportunities we have to perform longer tasks. We evaluate this percentage by sending tasks of different duration ranging from 1 minute to 35 minutes.

We note the probability of success of a task performed by nodes as $P_{Ts}(s_{c,T})$. We model MEC task offload as T_{off} and its duration $d_{T_{off}}$. We also note n_s the number of sets formed. We recall that $s_{c,T}$ the set of nodes located in cell $c \in C$ at time T. Thus, this section evaluates the practical feasibility of offloading tasks on such datasets by devising different scenarios. Figure 3.14 represents the results obtained for each of the datasets. We get a percentage of task success according to task duration and specified scenario.



(a) 1 km^2

(b) 4 km^2

(c) 16 km^2

(d) 64 km^2

Figure 3.13: Cell size variation in Rio from 4:00 am to 7:00 am.

3.5.1 Case 1: Percentage of successful tasks when at least one node is present in the set

In this first case, we assume that the MEC orchestrator sends a task T_{off} to all sets. A task will be well-executed if at least one node of the set is present during the required task's execution time ($d_{T_{off}}$). In cell c, at time T, the are n_s sets of nodes in the cell. Each set s_n, $n \in [1, s_n]$ has a hit probability of $P_{hit}(s_{n,c,T})$. This scenario induces a binary hit probability ($P_{hit}(s_{n,c,T}) = 1$ or $P_{hit}(s_{n,c,T}) = 0$). Therefore, we analyze the presence or absence of nodes for given tasks. Thus, we obtain a percentage of completed tasks according to a given duration. For a given $d_{T_{off}}$, the percentage of completed tasks is given by:

$$\frac{1}{n_s} \sum_{n_s=1}^{n_s} \mathcal{P}_{hit}(s_{n,c,T}) \times 100. \tag{3.4}$$

Based on Rome's scenario results, MEC can assign tasks requiring one minute of execution time with more than 80% of success. It is interesting to realize that around 35% of the nodes can complete a task of 10 minutes. On the other hand, for a task lasting more than 20 minutes, we have between 15 to 20% of the nodes capable of performing it successfully. Our experience shows that a minority of nodes (1%) can execute tasks of 200 minutes. For the city of Rio, the results are even more significant than Rome's. We have 95 of 1-minute tasks well performed by nodes present in each set. More than 45% of 10-minute tasks were process correctly. We found that a minority of tasks (3%) can perform up to 550 minutes.

In conclusion, tasks having a duration between one to ten minutes have a high success rate. In the formed sets, nodes are persistent enough to perform these tasks. Indeed, if we select a node at random, almost one out of two nodes can perform tasks lasting 6-8 minutes. Moreover, there is 1% of the tasks can be well performed for 24 hours. If there are only a few, it is still appealing to have the possibility to perform a task for an extended period. In this scenario, vehicular datasets provide a more significant percentage of tasks achieved with various task sizes, particularly the Rio dataset.

3.5.2 Case 2: Percentage of successful tasks when sets of nodes stay together as formed.

In this second case, we consider that a task is well-executed if none of the set's nodes left. This scenario also induces a binary hit probability. This scenario is the opposite of the

first one. We had assumed in the earlier one that as long as there is a node in the set, it is a successful task, whereas here if just one node leaves the set to consider the task lost. Generally, this scenario results in a lower percentage of nodes for task completion. It is a result that we expected due to the high mobility of nodes. A node will be more inclined to leave the set in a set of nodes, leading to a lower probability of task completion. It enables us to investigate how well a set maintains its formation state. In other words, it allows us to observe the cohesion of sets. We find the percentage of a successful task according to the given task duration.

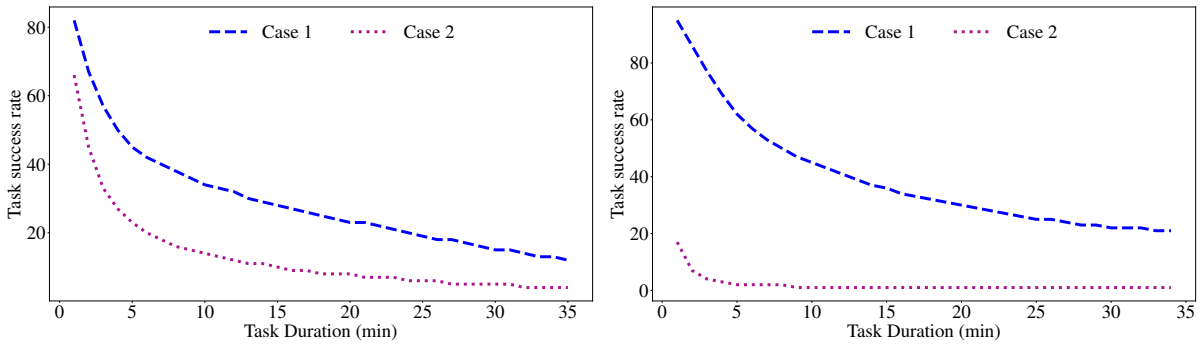
For Rome, the cohesion is maintained; the curve shape is very similar to the first scenario, with fewer tasks success. We have 66% of the 1-minute task's duration successfully perform and 14% of success for 10-minute tasks. For a task of 15 minutes, we get 10% of tasks successfully carried out. Above a task duration of 20 minutes, the percentage of successful tasks is low. It induces that the persistence of the set of nodes as formed in the initial state remain weakly maintained above a task duration of 20 minutes. So it is much less convenient to offload a task that requires all nodes to be available. Thus, if MEC needs to parallelize a task with sub-tasks with dependencies, it is recommended to execute short tasks to optimize success. We observed that less than 20% of the nodes accomplish their 1-minute tasks for Rio. It implies that these datasets have nodes much more mobile than Rome; the cohesion is therefore small.

We observe that the vehicular datasets are suitable for offloading tasks. Mobile nodes realize a large percentage and varying tasks sizes as we observe. However, the nodes within sets are very mobile, as pointed out in scenario two. We observe a greater cohesion for the Rome taxi dataset than the Rio one.

Main outcomes: *We observe that vehicular datasets are suitable for offloading tasks. Mobile nodes perform a large percentage of tasks of varying sizes. However, the nodes within sets are very mobile, as pointed out in scenario two. We observe a greater cohesion for the Rome taxi dataset than the Rio one.*

3.6 Conclusion

We proposed an offloading recommendation-based system following task success probability. The MEC is aware set of nodes that can accomplish tasks at a specific location. For this, we have explored temporal characteristics of extending MEC capabilities beyond the edge using mobile nodes. Our work focuses on the formation of available computing sets and their spatial and temporal persistence. Our results show that sets persistence follows a pattern depending on traffic density and the time of day. The evolution of per-



(a) Rome (b) Rio

Figure 3.14: Task well-executed for each dataset and scenario.

sistence analysis suggests that peri-urban areas, which are transit zones, are well suited to short discharges (low persistence). Although the data analyzed are scenario-specific, we suggest dissociating high and low affluence regions to have more accurate mapping. Furthermore, we believe that the co-localization of nodes is very relevant information to plan the placement of MEC to use they nodes or conversely. In the next chapter, we continue leveraging mobile devices by exploring the resources that a cell can provide to offload tasks.

—It is our attitude at the beginning of a difficult task which, more than anything else, will affect its successful outcome —

William, James

4

Offloading computing tasks beyond the edge in a cell

By leveraging available computing resources at end-user devices, operators can alleviate their servers while reducing costs. As we already know, a problem in such a system is that devices may move or leave the network before completing a task. In this chapter, we investigate the impact of the mobility of the nodes on the efficiency of the offloading system per-cell basis. Using a real dataset of thousands of people in the streets of Paris, we quantify the intensity of achievable task execution. By varying the tasks' resource requirements and the deadlines for nodes to conclude tasks on behalf of the MEC.

The outburst of mobile devices comes with an equally massive amount of data to process. As a result, operators must ensure a high quality of service to end-users and the availability of resources. Residing at the edge of the network, MEC solves cloud computing latency and traffic congestion problems associated with transmitting large amounts of data outside the core network [2, 3, 4, 5, 6]. However, large-scale deployment and high costs make it difficult to reap the benefits of MEC [7].

For a long time, mobile devices have been considered the weakest link of the network infrastructure. However, the number of mobile devices is increasing continuously. Ad-

ditionally, as the years go by, those devices tend to become more powerful and possess increased computing, storing, and sensing capacities. Besides, they are rarely used to their full capabilities and are often left idle [8, 9]. Given their widespread use, it drives us to consider them as a valuable resource to leverage. Thanks to their ubiquity, they compose a resource that does not imply any deployment cost for operators. In this work, we consider an unusual approach. It entails offloading computing activities from MEC servers to mobile devices, allowing computation *beyond the edge*. Using mobile devices as a workstation not only allows relieving some of the network infrastructure's load, but it can also boost the MEC server's availability.

Although assigning computing tasks to devices beyond the edge looks promising, it raises many challenges as the corresponding computing resources are volatile due to the inherent mobility of the nodes. In this chapter, we assess the impact of mobility on the ability of the end-nodes to offload tasks from the MEC. To this end, we tile the geographical area with cells, which we consider as the zone where the MEC can allocate tasks. In other words, we seek *to quantify the resources that a cell can provide to perform a MEC task*. Based on the resources a cell can provide, it is possible to determine which cell to offload tasks to at a given time.

We investigate how much the latent computing resources of end-user's devices could resolve tasks that the MEC's servers have offloaded.In this chapter, our objective is to estimate how many tasks assigned by the MEC's server nodes can undertake. We characterize a task as a *duration*, i.e., the time it takes for a mobile node to complete the task's execution. We further consider that a mobile device must complete the task within some time window, which we call *completion delay*. To assess the success of task execution, we compare the task duration with the time the node is present in the cell, allowing us to obtain the *potential* number of nodes that would be able to execute the task. We also evaluate the influence of the completion delay on task execution. In our study, we use real mobility traces to assess available offloading resources.

We make several observations. Firstly, the longer the completion delay, the greater the chances for task completion. Continuously extending the completion delay, on the other hand, is pointless after a certain amount, and the benefits become negligible. Depending on how quickly the MEC needs tasks to be processed, it can identify the maximum number of specific tasks that nodes can undertake. Furthermore, we show how the capacity of cells varies significantly depending on whether the area is residential or not. We also indicate the average number of tasks a node can perform based on its cumulative presence and allotted time. As a result, it enables us to extract the total capacity of a given cell and the resources that a single node can provide.

As a summary, the contributions of this chapter are:

- **Proposal of a metric to characterize tasks that nodes can undertake.** We exploit the cumulative presence time for understanding the possibilities of using end-user devices to extend MEC's capacity. Therefore, we propose calculating the cumulative presence time of users in their cells to investigate the effects of device mobility on offloaded task completion.

- **Evaluation of offloading potential in a pedestrian scenario.** We assess our analyses using a dataset collected in a pedestrian scenario by a major European mobile operator's core network. It describes the mobile traffic generated by all subscribers in the north area of Paris [13]. We have 708 cells and 14,727,959 users.

- **Measurement of offloadable tasks in the function of their requirements.** We assess the practical viability of offloading tasks on such datasets using different task sizes. We distribute tasks that require 5 to 35 minutes of execution time to all nodes. As a result, we obtain the number of feasible tasks.

- **Analysis of the impact of the completion delay.** We vary the completion delay and see how many resources we can acquire on behalf of the MEC. Thus, by identifying the percentage of tasks that any nodes can handle in a given cell, MEC can estimate whether the cell is suitable for offloading and whether the completion time is worth the additional tasks' opportunities. We finally identify the percentage of cells with a specific increase in task feasibility by using completion delay.

The chapter is structured as follows. In Section 4.1, we introduce the cumulative presence time of nodes in a cell as a metric for estimating a cell's offloading potential. We also present the completion delay. In Section 4.3, we investigate the number of nodes that may carry out a given task successfully by using a real-world cellular dataset. We vary the completion delay to observe its impact on the offloading potential. We analyze the capabilities of cells to perform tasks on a larger scale over a day. Finally, we discuss the benefit gained by completion delays. We conclude the chapter in Section 4.5.

4.1 Offloading strategy and assumptions

In this section, we introduce our methodology for estimating a cell's offloading potential. First, we present our task allocation process from the MEC to end-nodes. End-nodes in a cell are the critical element of our study. They represent *computational resources*,

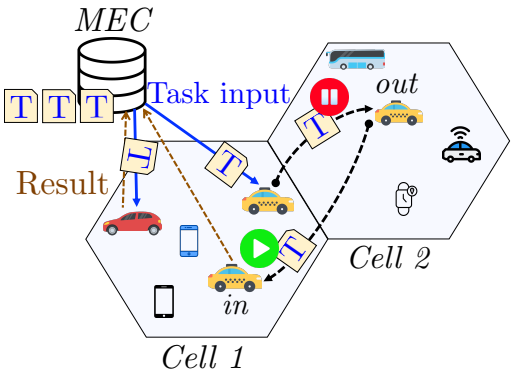
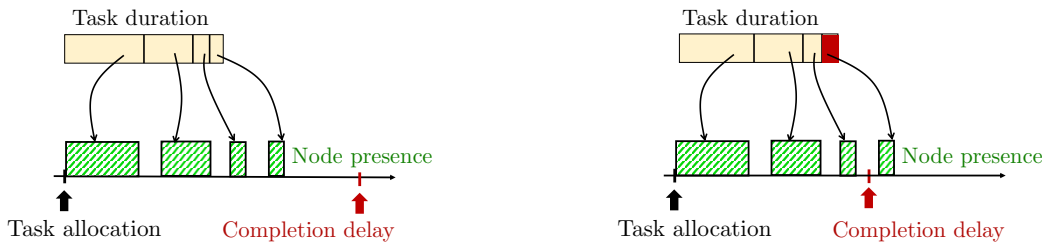


Figure 4.1: Offload computing tasks to mobile devices without losing the task processing when they leave the cell.

i.e., a measure to help evaluate the computing load that a cell can handle. It is thus necessary to understand the system's dynamics so that the MEC orchestrator can have an idea in which cell it can offload tasks. Next, we explain the cumulative presence time and completion delay notions. Finally, we endorse our proposal through a theoretical explanation.

4.1.1 Task allocation process

We consider a scenario where an operator seeks to reduce MEC workload. First, it needs to know in which cells MEC can offload. More specifically, we propose a scheme where the MEC offloads tasks to end-nodes. We model the allocation process of computing tasks to end-nodes in two steps. First and foremost, we assume that the MEC can only offload a given task to end-nodes taking part in its cell. Indeed, this strategy excludes the communication cost and management overheads among base stations. Moreover, it allows the MEC server to have complete control over its tasks management and the available resources at a given time. Secondly, the end-nodes are assumed to process an offloaded task inside the cell managed by the offloader's MEC. Suppose such a node goes out of its offloader's cell. In that case, the mobile node suspends the task to make its computing resources available to the other MEC servers. When the node returns to the cell that initially assigned the task, it resumes execution to give the result back to the original MEC server, as depicted in Figure 4.1. The second case involves the node remaining in the cell (Figure 4.1). It completes the task and returns it to the MEC who assigned it. If the node never returns to the cell, a third-party cloud, another node, or even the MEC itself will handle the task. At this point, end-nodes' continuous presence in the cell is unnecessary to complete the assigned task's execution as long as the node returns to the



(a) Successful task execution.

(b) Failed task execution.

Figure 4.2: Illustration of the presence time and completion delay.

original cell in the allotted time.

4.1.2 Cumulative presence time and completion delay

We consider the cumulative presence of end-nodes in our model. It allows compensating for the end-nodes mobility characteristic and having more computation time for a task. Nevertheless, we require that end-nodes complete the task assigned by the MEC within a given time window. We call this time window the *completion delay*. To ensure that a node can accomplish a task before the completion delay, we compare the task duration to the time the node spends within the cell. The node's cumulative presence time in the cell before the completion delay expires must be equal to the task's execution time.

Figure 4.2 highlights our hypothesis that the use of cumulative presence time enhances task accomplishment despite node mobility. A node receives a task of a specific duration. The boxes represent the presence time of the node in a cell, and the blanks indicate its absence. Besides, Figure 4.2 puts forward the influence of completion delay. In Figure 4.2a, the completion time is significant. Hence, the cumulative presence of the end-node allows a well execution of the task and returns it to the MEC. On the other hand, in Figure 4.2b, the completion time is shorter. The task is, therefore, not completed despite the use of the cumulative presence of end-node. In that case, a remote cloud or the MEC itself will perform the task.

4.1.3 Formalization

We define a task as the basic unit of work that contains instructions to accomplish a specific functionality. We note T the time it takes to execute a task and t when the task is assigned to a mobile node n. Thus, the node completes the process only if its

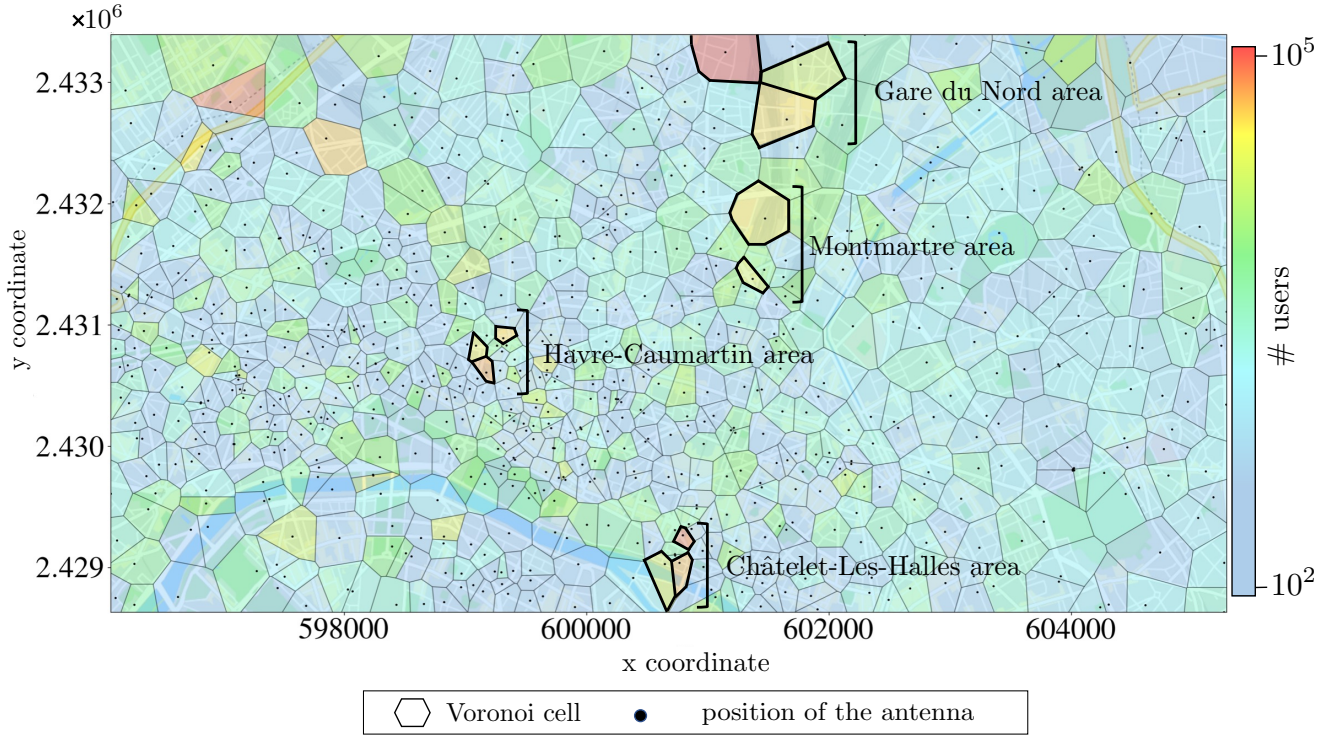


Figure 4.3: Number of user in cells during 24h.

cumulative time of presence in the cell is at least T. We consider a continuous-time and define $\hat{n}(t) = 1$ if the node is present in the cell at the time t and $\hat{n}(t) = 0$, otherwise. We thus define the cumulative time of presence of node n over period. We note the completion delay as d_c.

Thus, we consider that node n completes the task if:

$$T \leq \int_{t}^{t+d_c} \hat{n}(u)\ du. \tag{4.1}$$

4.2 Pedestrian dataset

We use a cellular dataset collected in a major European mobile operator's core network to analyze pedestrian scenarios. The dataset describes the mobile traffic generated by all subscribers in the north of Paris [13]. First, we outline our dataset. Then, we present general analysis such as the average frequency of return of nodes in a cell and the average presence time of node in a cell.

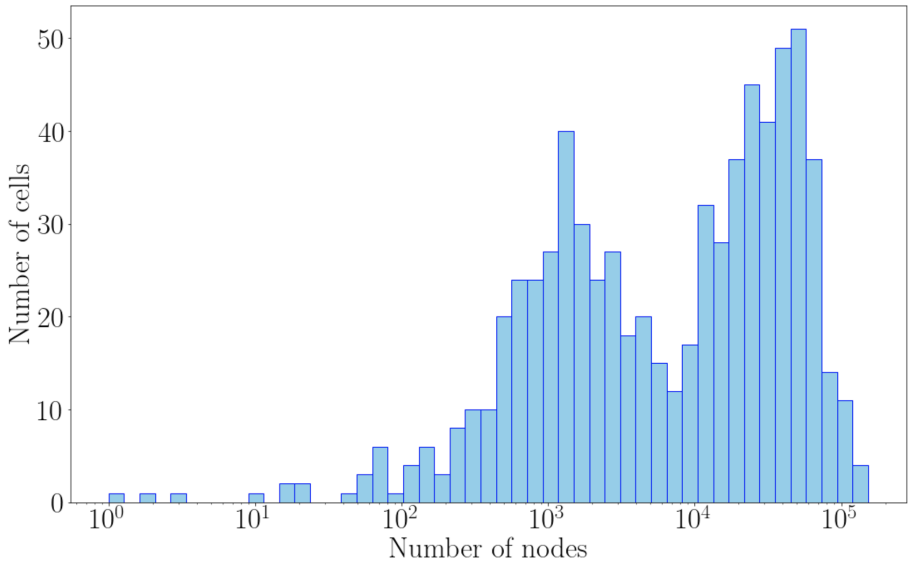


Figure 4.4: Cell distribution of nodes number.

4.2.1 Number of users in cells

First and foremost, to conduct our analysis, we compute the Voronoi tessellation from the list of the antennas' coordinates. The distribution of antennas reflects the city's structural and human activity features; it creates an effective spatial partition. Figure 4.3 depicts the Voronoi tessellation of the studied area. We have 708 cells that we explore during a 24-hour window on a weekday using a granularity of five minutes. There are 14,727,959 users in all cells for the selected period. The color of each cell of the diagram varies with the number of nodes in cells. From the blue for the smallest node number to red for the largest and passing through green and yellow. Throughout the day, the lowest number of nodes found in a cell is 11, and the highest number is 152,775.

At first sight, we notice a very homogeneous distribution of the number of nodes in Paris. It is particularly noticeable in the North-East of Paris. Nevertheless, there are places where the population is more concentrated than others. For example, we observe between 40,000 and 80,000 users in the cells along the Seine river. We notice a few cells containing between 112,387 and 128,057 nodes, represented in orange-yellow. These are places representing a point of interest. Among the cells containing more users, we have the railway areas such as Gare du Nord. These are the cells located at the top of the map on the left side. Right below, we have a dense cell located near the Montmartre area, one of the most famous Parisian neighborhoods. Among the small cells, but with many nodes, we have one located near the Seine river. These are cells found in the historical center of Paris, near the famous Notre-Dame de Paris cathedral and Châtelet-Les-Halles district.

Many tourists and locals are attracted there by the numerous shops, pubs, theaters, cinemas. We also see a cell located in the Havre-Caumartin district, along with the very dense cells. It is a very dynamic area starting at the Place Saint-Augustin and imposing church and ending at the level of the Printemps and Galeries Lafayette stores, passing by the Opéra Garnier. The activity is intense, and bankers, shoppers, cinema-goers, and tourists constantly cross each other.

Given the homogeneity of nodes number in cells of Figure 4.3, we express the numbers of the cells according to the number of users in Figure 4.4. Therefore, it allows a better understanding of the distribution of users. By looking at the histogram, we can tell that out of 708 cells, very few cells (2.5%) have less than 100 nodes. We find around 19% of cells that have between 100 and 1,000 nodes. 28% of the cells have between 1,000 and 10,000 nodes. The majority of cells have between 10,000 and 100,000 nodes. We have a majority of the cells with such an appealing percentage of nodes. Specifically, 51 cells are the most significant number of cells containing 735,633 users. Finally, we have a minority of cells (2.11%) going beyond 100,000 nodes.

Main outcomes: The majority of cells have an attractive number of nodes over 10,000. It is particularly promising since the more nodes there are, the more they can work. Moreover, the cells with the highest nodes are primary points of interest, such as transportation facilities (train stations, subway stations), shopping malls, and historical and cultural places. The number of nodes present in a single day confirms that the intuition of using the plethora of mobile devices beyond the edge is promising.

4.2.2 Average frequency of nodes return

To enhance the use of mobile nodes, we are equally interested in knowing if the nodes tend to return to a given cell. Therefore, for each cell and its respective nodes, we analyze how often they come back. Thus, Figure 4.5 depicts the percentage of cells according to the average frequency of nodes return. We find that the nodes return on average more than once to their cells without exception. We have 20% of the cells with nodes that return an average of 1.5 times in the cell. It is the highest percentage of node returns, followed by 16.5% of the cells, with nodes returning an average of 2 times. Then we have 15.8% of the cells with nodes that return an average of 2.5 times and 14.8% for three returns, and finally, 10.0% of the cells with nodes that return 3.5 times. We have a considerable decrease in the percentage of cells (between 6% and 3% of the cells concerned), whose nodes come back more than four times. In a minority of cells, we have up to 13 returns of nodes to the cell in 24 hours.

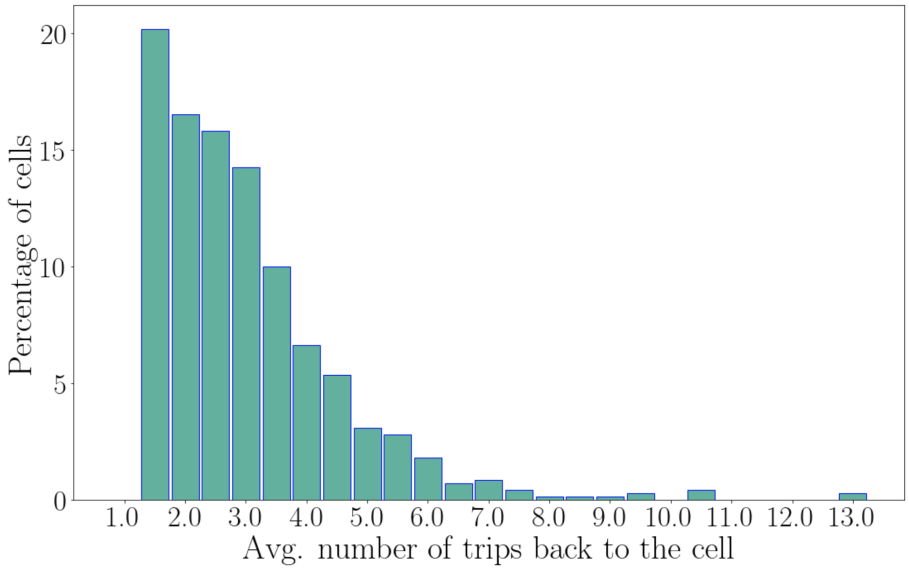


Figure 4.5: Average node return in cell.

Main outcomes: Overall we have 83.7% cells whose nodes return in the cell between 1.5 to 4 times in a day. We assume that these are the classic round trips of an ordinary user, going from his home, workplace, nearby shops, schools. Thus, according to our offloading strategy, the return of nodes to the cell is beneficial to expand MEC capacity.

4.2.3 Average presence time of node in a cell

Having many nodes and knowing that they come back into the cell is a positive indication that our idea is sustainable. Nevertheless, we go a step further by investigating whether nodes remain in a cell long enough so that the MEC can offload computing tasks to them. Thus, Figure 4.6 represents the average presence time of nodes in cells. We realize that many cells have a high average presence time. 72% of the cells have nodes whose average presence time is between 5 and 30 minutes. A task between 5 and 30 minutes is therefore achievable in a larger part of the cells. We have 17% of the cells whose average presence time of the nodes is between 30 and 50 minutes. Finally, we have 9% of the cells whose nodes have an average presence time of up to 100 minutes. These cells are useful for the execution of long tasks or several short ones. There are also 2% of the cells not represented on this figure whose nodes presence time reaches 300 minutes.

Main outcomes: The most common average presence time (52 cells) is 8 min, followed by 40 cells with an average presence time of 9 minutes. Thus, overall, we observe that the MEC has the opportunity to offload both short and long tasks in the northern area of Paris.

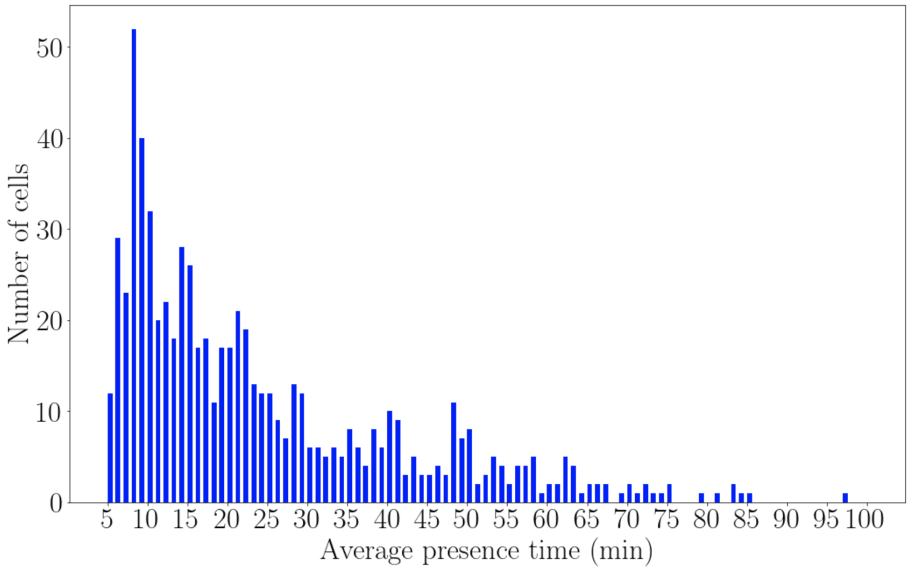


Figure 4.6: Distribution of the average presence time of nodes in cells.

4.3 Varying the completion delay

In this section, we investigate the task opportunities for MEC offloading. We also analyze the impact of the completion delay on task offloading opportunities. To visualize the effect of the completion delay step by step, we increase it from 10 to 120 minutes. We study cumulative presence time for two cells between 11 a.m and 2 p.m (i.e., 3 hours) for the sake of readability. We discretize time into 5-minute intervals. The cells are:

- Cell 1 is located in a residential neighborhood in the downtown district of Parmentier.
- Cell 2 is located near Gare du Nord, the largest railway station in Paris and a major transportation hub for both commuters and distant travelers.

Let us take a look at the number of users that each cell can offer. We have a total of 659 end-nodes for cell 1 and 13,571 end-nodes for cell 2 during the observation period. Thus, we observe a significant amount of resources for the MEC server to offload tasks. We strongly believe that there is a potential capacity to be exploited beyond the edge. Let us further examine how this capacity evolves with different tasks duration and by increasing completion delay.

In Figure 4.7, we plot the number of tasks successfully achieved while increasing the completion delay[1]. We compute each node's cumulative time presence in the cell and

[1]Note that the y-axis graduations are not identical to ensure clear readability of the curves.

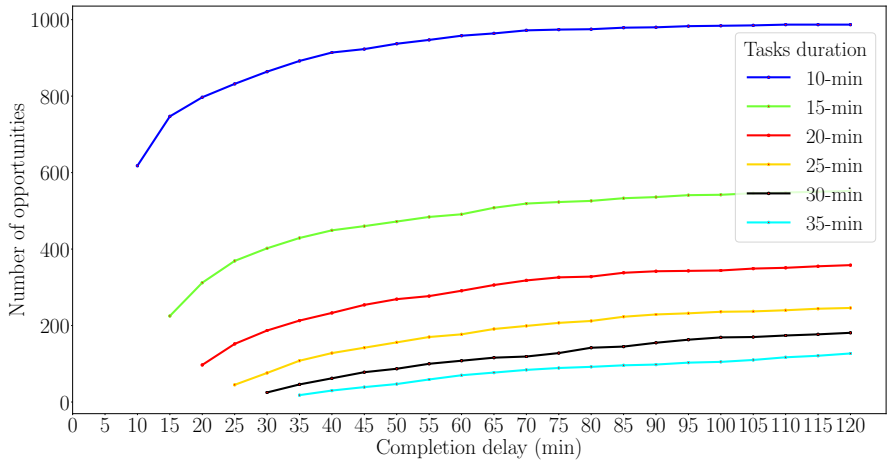
deduce, for a given task duration T, whether it can perform the task in the given period and how many such tasks it would be able to perform the task in the given period to accomplish. We consider tasks with duration T ranging from 10 to 35 minutes. We make the following three main observations.

Firstly, among the 659 end-nodes present in cell 1, almost all end-nodes can perform a 10-minute task. Indeed, for the shortest task duration, the MEC can offload 615 tasks. Regarding cell 2, there is also a massive opportunity for the MEC to offload tasks with the shortest task duration of 10 minutes. It represents 73% of the end-nodes, which is 9914 opportunities for the MEC offloading. Thus, we obtain task offloading opportunities even in a scenario using the shortest task duration (10 minutes) and completion delay (10 minutes). It confirms the approach showing that MEC can offload tasks to end-nodes.

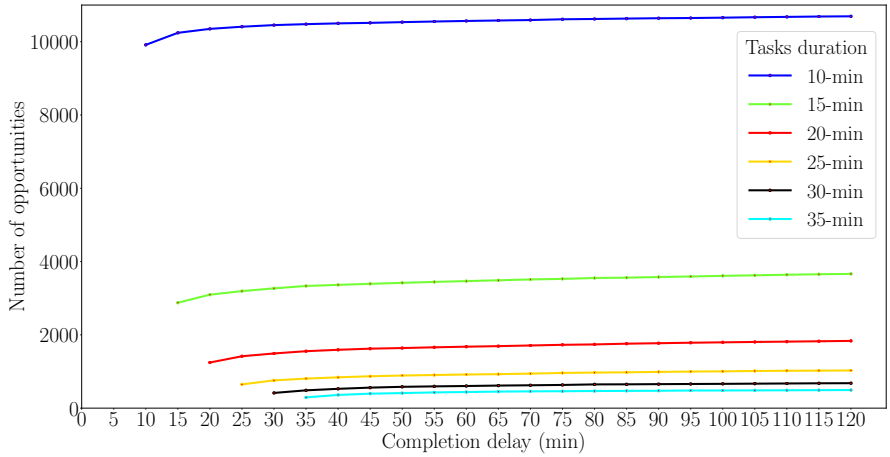
Secondly, increasing the completion delay continually improves the number of task offloading opportunities. Nevertheless, there are some discrepancies depending on the cells. We observe these discrepancies for $T = 10$ minutes. For example, in cell 1, we observe that it is possible to run between 615 to 987 tasks by increasing the completion delay value from 10 to 120 minutes. Thus, we get an additional 16% opportunities to perform the task. Moreover, we reach a certain time plateau at a given time where the completion time has a minor impact. This time plateau is visible after 105 minutes for cell 1. For cell 2, a significant number of end-nodes can handle a 10-minute task. There are between 9,914 and 10,693 task offloading opportunities—nevertheless, completion delay has a minor impact of only 3% additional possibilities. Thus, it is clear that the increase in completion time has a minor effect. We also notice the appearance of a "plateau" at a certain point in time. It corresponds to when the completion delay's impact becomes negligible, and the number of opportunities becomes almost constant.

Thirdly, when the task's duration increases, there is a sharp drop in task offloading opportunities for the two cells. For the most extended completion delay (120 minutes) and tasks between 15 and 35 minutes, we have between 127 to 358 opportunities to offload tasks in cell 1. For cell 2, the MEC can offload between 494 and 3,663 for tasks of 15 to 30 minutes. We notice that cell 2 has fewer options for tasks up to 35 minutes compared to its initial capacities. However, 494 possibilities to perform a 35-minute task are still very appealing. It represents 17,290 computing minutes, alternatively 12 days of computing available for the MEC in 3 hours. On the whole, we observe a decrease in the task offloading opportunities for the MEC for 15 to 30 minutes tasks.

MEC has plenty of opportunities to offload the tasks. By using the completion time, opportunities become more abundant. It is more evident in cell 1 than in cell 2. We believe that it is due to the nature of the two locations. The first cell represents a residential



(a) Cell 1.



(b) Cell 2.

Figure 4.7: Impact of varying task completion delay. The plots show the number of task offloading opportunities, for the two cells, for tasks of duration ranging from 10 to 35 min, and with increasing completion delay (up to 120 min).

area, where we expect low mobility. Thus, the increase of the completion delay allows a considerable gain of tasks. As mentioned above, the other cell is a cell with high traffic because of the many transportation facilities available, which means that the mobility of these end-nodes is high. Therefore, the increase in completion time does not obtain as much gain as in cell 1. We see in Figure 4.7, the appearance of the "plateau" is sooner in cell 2.

Main outcomes: In conclusion, the impact of the completion delay is variable depending on the kind of cells. Hence, there is a trade-off between the completion delay tolerance and the number of opportunities gained depending on MEC tasks duration and tolerance of tasks returns.

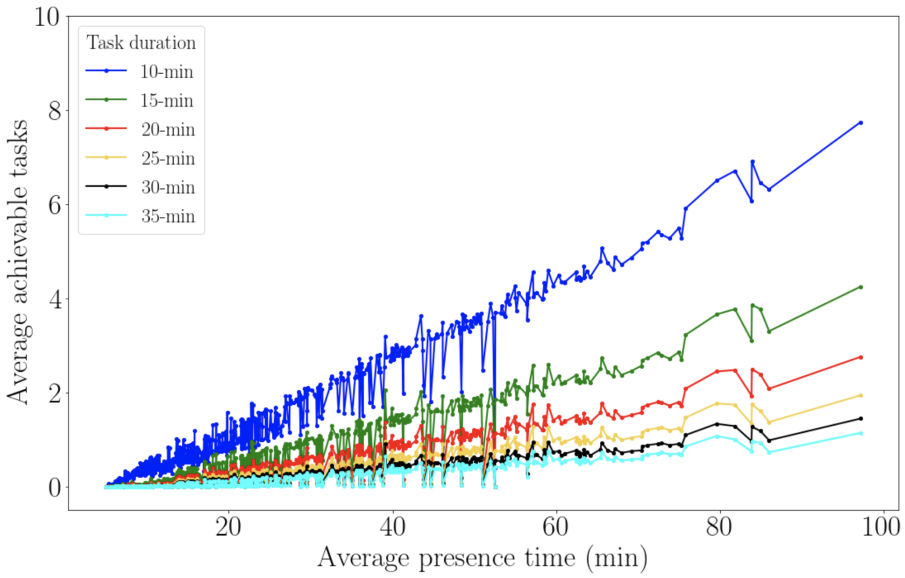


Figure 4.8: Average number of tasks with a zero completion tolerance delay.

4.4 Capacities provided by a node in a cell

We aim to quantify the resources that a cell can provide to perform a MEC task. We thus extend our analysis to the 708 cells presented in Section 4.2. In this section, we investigate the average task computation that a node can offer over 24 hours. For that, we first calculate users' presence time in each cell. Then, based on the average presence time of end-nodes in a given cell and the task duration to be executed, we determine the average number of tasks that a node can perform. Next, we use extreme values of completion delay: one with a zero-tolerance and the other with a maximum tolerance of 24 hours. It enables us to identify two bounds, one corresponding to the number of resources provided at a given moment and the second corresponding to the maximum number of resources provided in a given period. We vary tasks' duration from 10 to 35 minutes in both cases. Finally, we discuss the gain obtained by using a completion delay.

4.4.1 Case 1: Average number of tasks with a zero completion tolerance delay

We assume that the MEC orchestrator sends a task to end-nodes in a specific cell. A task is well-executed if end-nodes are present during the required task's execution time. In this case, the tolerance for completion delay is zero, implying that the execution time is equal to the task size. End-nodes must remain in the cell to complete the task and return

it to the MEC. Otherwise, the work is considered incomplete. It is either reallocated to another node or supervised by the MEC.

Figure 4.8 represents the average number of achievable tasks according to the average presence time of end-nodes in the cell. Each point on the curve reflects the 708 cells evaluated along with average tasks feasible. One of the first observations that stand out is that most cells have an average presence time from 5 to 60 minutes. There are some cells where end-nodes have an average presence time between 60 and 100 minutes. Finally, we find a minority of cells with an average presence time of up to 100 minutes.

For cells whose average presence time is between 5 and 60 minutes, the end-nodes perform an average between 0.44 to 4.5, respectively, for a 10-minute task. However, if the MEC wants to ensure that all end-nodes present perform at least one 10-minute task, it must select a cell with an average presence time of 17 minutes at least. In the same way, for a more extended task, MEC has to choose a cell with a longer average presence time. For example, for a 20-minute task, the MEC has to select a cell with an average time presence of 42 minutes to ensure that any end-nodes can complete it. On the other hand, cells with end-nodes having an average presence time of more than 60 minutes can perform an average of 4.5 to 8.4 10-minute tasks.

Regardless of the completion time, it is worth noting that a node's ability to accomplish short tasks is more significant than large ones. For example, for the longest average presence time of 100 minutes, a node can achieve an average of 8.40 10-minute tasks. Likewise, for tasks between 15 and 35 minutes, the node can accomplish an average of 4.62, 2.96, 2.15, 1.45, and 1.25 tasks, respectively. The end-nodes perform an average of 0.05 to 4.5 10-minute tasks in the case of most cells with an average presence time between 5 and 60 minutes.

Main outcomes: The overview of the distribution of 708 cells based on their average time presence and what each node contributes as a mobile workstation allows the MEC to better understand beyond the edge. As a result, depending on the number of tasks the MEC needs to offload, it can select the suitable cell. Therefore, even when the completion time is zero, we have multiple offloading potentials for the MEC.

4.4.2 Case 2: Average number of tasks with a 24-hour completion tolerance delay

In this case, the tolerance for completion delay is the maximum window equivalent to a 24-hour period—it implies that the end-nodes have to return the task within this completion delay. Naturally, end-nodes can take on a new task if they complete it before the

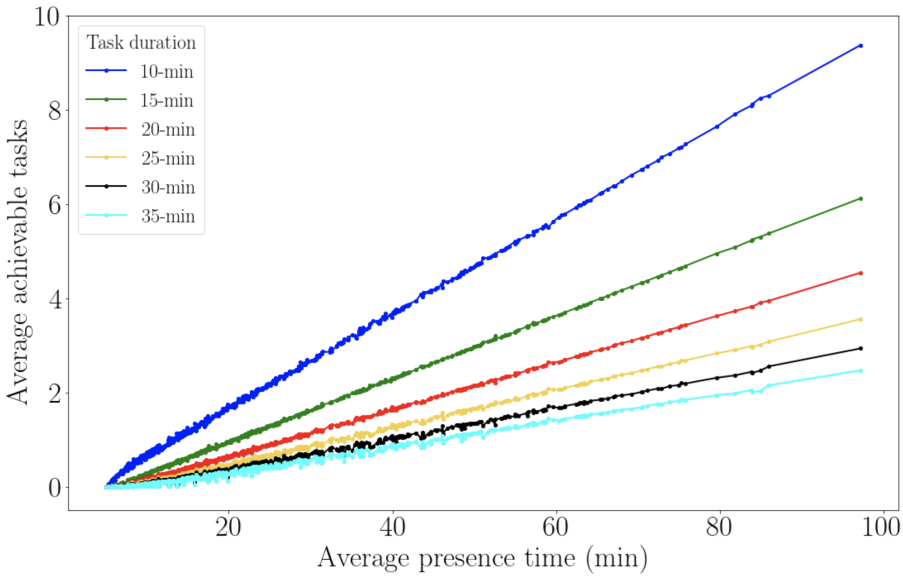


Figure 4.9: Average number of tasks with a 24-hour completion tolerance delay.

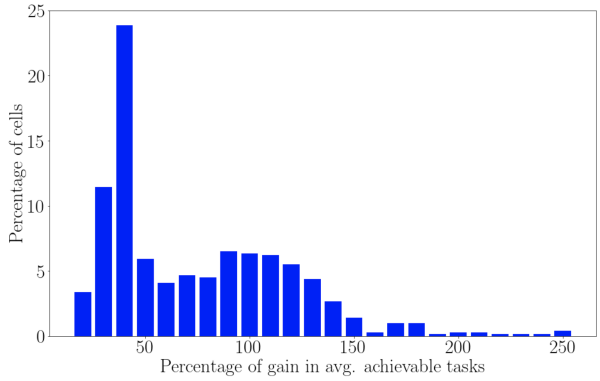
completion delay. On the other hand, if the end-nodes never return to the cell, the task is lost.

Figure 4.9 depicts the average number of achievable tasks according to the cumulative time presence of end-nodes in the cell. First, we notice that curves are much smoother. In contrast to Figure 4.8, we have a majority of cells whose average presence time is between 5 and 70 minutes instead of 5 and 60 minutes. Nevertheless, we still find a minority of cells whose average presence time of the end-nodes goes up to 316 minutes.

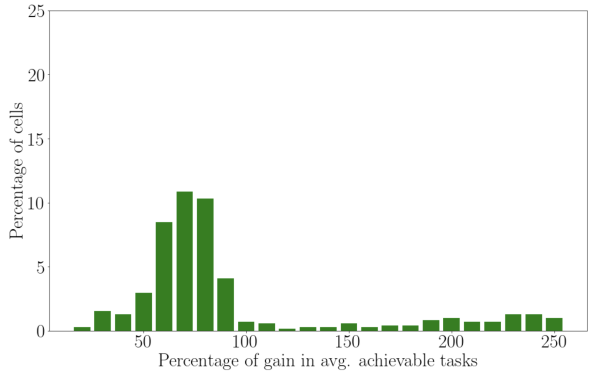
We observe an almost linear growth of the average number of feasible tasks according to the average presence time with completion delay. Indeed, if we consider the cell with an average presence time of 50 minutes, the end-nodes can execute 4.6 tasks of 10 minutes. If we pick another example, an average presence time of 100 minutes, the end-nodes can accomplish 9.8 tasks. Such linearity allows the MEC to foresee the number of achievable tasks.

Furthermore, we see a significant increase in achievable tasks by having a 24-hour delay completion. For example, for 10-minute tasks, in most cells with an average presence time between 5 and 60 minutes, the end-nodes perform an average of 0.72 to 5.7 tasks. Thus, end-nodes carry out an average of 5.7 to 10.3 10-minute tasks for cells with an average presence time between 60 and 100 minutes.

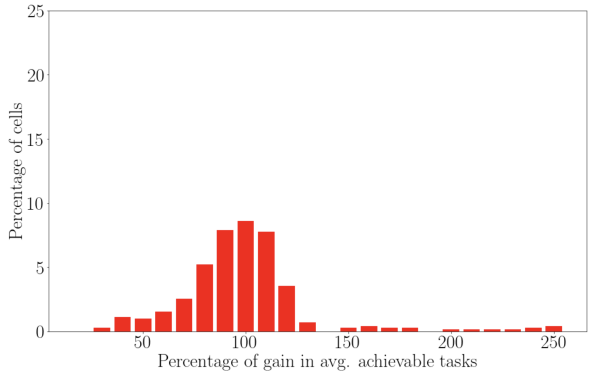
Main outcomes: The overall impact of the completion delay is positive. The high tolerance from MEC regarding the completion delay brings the closest to the maximum computational bound that a node can provide. Thus, it allows increasing the number of



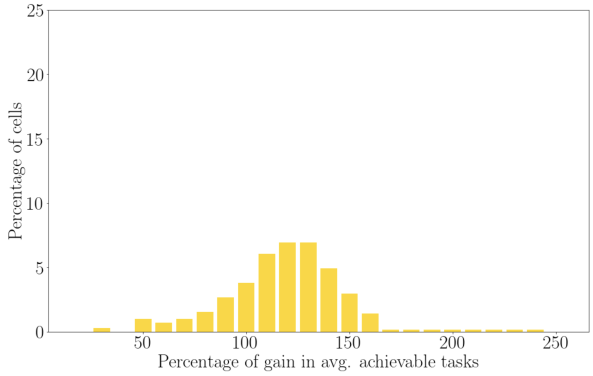
(a) 10 minute task gains.



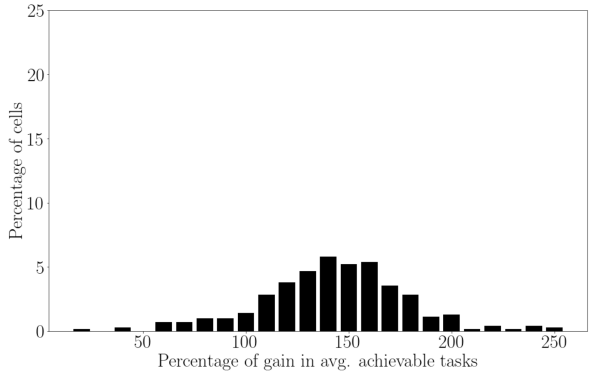
(b) 15 minute task gains.



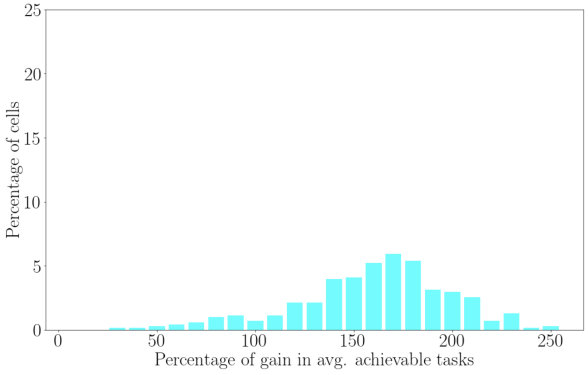
(c) 20 minute task gains.



(d) 25 minute task gains.



(e) 30 minute task gains.



(f) 35 minute task gains.

Figure 4.10: Additional tasks gain opportunities obtained by using completion delay.

opportunities to offload tasks from the MEC.

4.4.3 The gain derived from the completion delay

To improve the impact of completion time analysis, we analyze the percentage of additional tasks obtained in case 2 compared to case 1. Thus, Figure 4.10 represents the percentage of cells that obtain a percentage of feasible task gain using the completion delay. Each sub-figure represents the gains obtained for different task sizes ranging from 10 to 35 minutes. We observe that the potential gains obtained are very high, exceeding 100% for all task sizes. However, we only present the gains obtained up to 250% of additional achievable tasks because most cells are at this level, and only a minority can achieve gains of 1000%.

Overall, the longer the task, the less the cells are affected by tasks execution gain. For example, for 10-minute tasks, the highest cell percentage is 24% which acquires a 40% tasks execution gain. On the other hand, we have 70% of 15-minutes tasks earning 10.8% of the cells. Thus, we have 7.9% of the cells with a 90% tasks execution gain. Finally, we have 6.9% and 5.7% of the cells with a gain of 120% and 140%, respectively, for 25 and 30-minute tasks. So, it confirms that the longer the task, the fewer the number of cells acquiring task execution opportunities. However, this analysis puts forward another element. Despite the decrease in the number of cells exhibiting a gain in tasks execution opportunities, the gains for the remaining cells are considerable. We notice that for the tasks of 15 minutes, the significant gains are between 50% and 90%, those of 20 minutes between 70% and 110%. We have between 100% and 150% for 25-minutes tasks and between 110% and 200% for 30 minutes. Thus, the higher the gains, the duration of the task.

For task gains of 10 minutes, most cells (89%) gain between 30% and 130% of additional tasks using the completion time. We notice that 24% of the cells are concerned with obtaining a gain of 40% tasks of an extra 10 minutes. It is the highest percentage of cells. It is noteworthy that we find a similar pattern for task gains of 15 and 20 minutes. Moreover, the same holds for the 25 and 30-minutes tasks. Nevertheless, we notice that the gains for the 15 and 20-minutes tasks are concentrated around 50% to 120%. Then we observe an apparent decrease beyond these values. For tasks of 25 to 30 minutes, the range of gains is much wider, from 50% to 200%.

Main outcomes: The percentage of additional tasks performed using the completion delay is significant. We obtained gains of more than 100% for all task sizes. Two significant insights are that the longer the task, the less the cells are affected by the task completion

gain. Secondly, the higher the gains, the longer the task duration. To conclude, the cumulative presence of users and the completion delay allow MEC to have systematic opportunities to offload these tasks.

4.5 Conclusion

Our approach consists of using end-nodes on behalf of the MEC to multiply its computing possibilities. We adopt a data-driven approach to evaluate the potential of offloading computing tasks on end nodes. Our work focuses on the characterization of cells capacities. For this purpose, we analyzed the impact of the number of possibilities to offload computational tasks by varying the completion delay. Our results show that end-nodes can achieve a considerable quantity of different size tasks in the north area of Paris. Moreover, depending on the delay completion tolerance, MEC can grab significant opportunities. Finally, this chapter concludes our contributions to exploring the possibility of using mobile devices on behalf of the MEC. The next chapter concludes this dissertation.

—Find the confidence in whatever way you can to just keep moving onto the next page. The only way you will finish projects and get better is to keep moving forward—

Charles, Forsman

5

Conclusion and futures perspectives

This chapter summaries research works presented in this manuscript. In addition, it highlights some directions to pursue in future works in this area.

5.1 Conclusion

The ubiquity of mobile devices and the desire to improve the user experience is resulting in sophisticated applications, many of which require significant computing power on user devices. The completion of those applications requires intensive computations and short delay constraints. One of the technical approaches attracting considerable attention to improve the efficiency of cloud computing is MEC. On the one hand, multi-access edge computing shifts cloud services to the radio access network to bring computing capabilities closer to mobile users. On the other hand, MEC offers much more limited resources and, in some situations, may not scale with the number of tasks. One appealing idea is to extend the capacity of the MEC by leveraging the available resources at end-user devices.

In this thesis, on the one hand, we have exposed the possibility of using mobile devices to multiply the computational possibilities. Indeed, through the formation of available computational groups, their spatio-temporal persistence, and by varying the size of the

cells, we affirmed the possibility of offloading tasks from the MEC. Hence, based on our datasets, it is possible to extend MEC's capabilities beyond the edge. Moreover, our results show that the persistence time of groups follows a pattern depending on vehicular traffic density and the time of day. Analyzing persistence trends leads us to believe that peri-urban areas, transit areas, are suitable for short offloading and conversely for urban areas. Therefore, we suggest dissociating high and low affluence regions to have a more accurate mapping. Secondly, we quantify how much resources can be subtracted by sending tasks of various sizes and by varying the completion time allowed by the MEC. Our results show that each mobile device contributes as a mobile workstation allowing the MEC to offload its task. As a result, depending on the number of tasks the MEC needs to offload, it can select the suitable cell. Therefore, even without the completion time, we have multiple offloading potentials for the MEC. The feasibility of performing tasks increases significantly by using the completion. The high tolerance from MEC regarding the completion delay brings the closest to the maximum computational bound that a mobile device can provide.

To summarize, the key takeaways from this thesis are:

- Mobile devices are an untapped potential for MEC task execution. We illustrated this assertion with vehicular and pedestrian datasets.

- Analyzing the spatio-temporal dynamics of mobile devices using persistence metrics and presence time provides us with a better understanding of mobility, colocation between users, and attractive geographic locations.

- Refining and validating the use of mobile devices by offloading computing tasks in different scenarios (e.g., different cell size, different task size, different conditions to consider a task as well-executed).

5.2 Future perspectives

In this manuscript, we have presented our contributions to the MEC research area. Shortly, we plan to expand our current work since, at the moment, our work is only at the observation-deduction stage. Although this thesis proposes a unconventional research angle, we believe they can be extended to more comprehensive directions to enjoy mobile device capabilities fully. In the following, we list some directions that can be investigated in future works:

- **Prediction of the persistence time of a group of nodes.** We want to propose a metric/model to predict the persistence time of a group of nodes. Profiling the capacities offered by mobile devices can be a significant step for the proper task offloading process. Moreover, we also plan to investigate the persistence in more cities, different cells size over more extended periods to improve the analysis of the evolution of persistence.

- **Choosing the suitable cell in which offload a task.** The efficient use of mobile devices is essential to avoid wasting time when performing tasks. Therefore, choosing the appropriate locations and nodes to offload a task is essential for a more efficient MEC extension. We seek to work on a metric that considers the studied ones (persistence, presence time, completion delay) and other parameters, for example, the properties of the task, to allow better targeting of the locations or nodes where offloading tasks.

- **Estimation of storage, computing, and battery capacities.** Having a group of mobile devices co-located and the knowledge of the dynamics is not sufficient to have a complete estimation of the offloading possibilities. We have to pay attention to other parameters to evaluate and refine a potential offloading metric. That is why we will focus on the battery capacity of the devices and their available computing and storage in our future work. For example, mobile devices can remain together for a long time. However, if their batteries for executing a computing task are insufficient, the group is useless for offloading a task.

- **General modeling of available resources beyond the edge.** Our goal is to propose a holistic metric to give a more representative understanding of beyond-the-edge resources. This part of the work will bring together the different parts of our work together. Indeed, the co-location/persistence and estimation of storage, computing, and battery capacity to obtain complete cartography of resources beyond edge computing.

- **Pricing or Reward system for users.** The mobile devices belong to the owner. Thus, a pricing model for using resources plays a crucial role. The user must have a reason to let its mobile device exploit for MEC task computation.

- **Security issues.** Computation offloading means that the MEC sends information (i.e., the source code or the data) of its task to a third party, which can result in a significant security issue. How to be sure that no copy of the MEC data is made?

In addition, the edge server also needs to trust the user source code. Therefore, both the edge server and user need to build a trusted exchange. Indeed, how to be sure that the MEC is not overusing users' mobile devices? These questions are very challenging in the context of computing offloading. Consequently, it is imperative to consider security as an essential issue for actual mobile devices usage.

A

Résumé détaillé en Français

La bataille perpétuelle consiste à améliorer les performances en réduisant la latence et en approvisionnant des ressources tout en garantissant une qualité de service élevée aux utilisateurs finaux.

A.1 Introduction

Ces dernières années, les progrès technologiques des systèmes de communication sans fil ont induit des changements sociétaux. La popularité des appareils mobiles a façonné nos vies, les normes sociales influençant la manière dont nous percevons la nécessité d'accéder à davantage de ressources et ceux de manière instantanée. En effet, la demande d'une pléthore de services tels que le streaming vidéo, les jeux en ligne, les réseaux sociaux, la banque, la santé et les achats en ligne suscite un intérêt croissant. L'accès omniprésent à ces services nécessite une continuité de service, une mobilité sans faille, une efficacité accrue des ressources ainsi qu'une faible consommation d'énergie. Par ailleurs, l'extension de la couverture cellulaire mobile, l'utilisation massive d'appareils mobiles et les abonnements au haut débit mobile génèrent un trafic de données colossal. Ce trafic pose des défis aux opérateurs mondiaux en affectant les réseaux mobiles et sans fil, en plus des con-

traintes d'accès omniprésent à fournir. De plus, la croissance exponentielle des appareils connectés reflète l'effet d'avalanche du volume croissant du trafic mobile et sans fil. Selon les prévisions de Cisco, il y aura 29,3 milliards d'appareils en réseau d'ici 2023 [1], face à laquelle les opérateurs doivent maintenir une haute qualité de service.

Pour répondre à ces exigences, le monde universitaire et l'industrie explorent de multiples technologies. La réponse à la question de savoir où nous pouvons trouver une meilleure disponibilité des ressources informatiques, ainsi que la fiabilité requise pour fournir une connectivité fluide pour les applications sans fil, repose inévitablement sur le cloud. Pour répondre à ces exigences, le monde universitaire et l'industrie explorent de multiples technologies. La réponse à la question de savoir où nous pouvons trouver une meilleure disponibilité des ressources informatiques, ainsi que la fiabilité requise pour fournir une connectivité fluide pour les applications sans fil, repose inévitablement sur le cloud. Le cloud est une approche ayant fait ses preuves pour étendre les ressources des appareils finaux à l'aide de datacenters. Par exemple, la pratique consistant à décharger des tâches de calculs informatiques vers le cloud est une solution bien connue. Cependant, la latence causée par la distance entre l'appareil final et l'infrastructure du cloud ainsi que les coûts liés à la traversée du réseau ne sont plus adaptés aux applications gourmandes en ressources. Cela conduit donc à l'adoption de solutions alternatives. Il ne fait aucun doute que ces solutions alternatives doivent introduire une infrastructure plus distribuée améliorant l'efficacité au niveau local en rapprochant les fonctionnalités de type cloud vers les appareils mobiles.

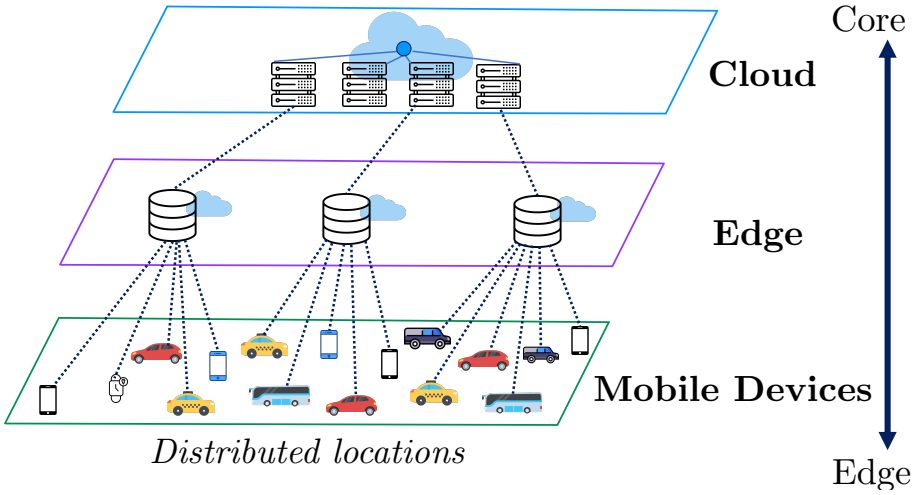


Figure A.1: Infrastructure hiérarchisée.

A.2 Les appareils mobiles : une ressource informatique supplémentaire

À première vue, l'utilisation de dispositifs mobiles pour étendre la capacité du MEC semble être une idée contre-intuitive. En effet, toutes les technologies mentionnées ci-dessus visent à surmonter les problèmes posés par la taille compacte des appareils mobiles (c'est-à-dire une batterie limitée et des ressources informatiques insuffisantes). Néanmoins, les dispositifs mobiles ont connu une évolution remarquable, passant de petits appareils aux capacités de ressources limitées à des accessoires essentiels de la vie quotidienne. Ils ont gagné en puissance et possèdent des capacités de calcul, de stockage et de détection accrues au fil des ans. En outre, ils deviennent de plus en plus abordables pour les consommateurs, ce qui facilite leurs achats. Mais surtout, ces derniers sont rarement utilisés au maximum de leurs capacités et restent souvent inactifs [8, 9].

Compte tenu de l'utilisation massive des appareils mobiles, cela nous pousse à les considérer comme une ressource précieuse à exploiter. De plus, ils constituent une ressource préexistante qui ne génère aucun coût de déploiement pour les opérateurs. Ainsi, nous considérons que les appareils mobiles sont un élément crucial à exploiter pour les opérateurs, notamment pour atténuer la quantité massive de données à traiter et le coût qu'elle engendre. La devise *"l'union fait la force"* nous amène à supposer qu'un appareil a peut-être une capacité insuffisante, mais qu'un milliard peut changer la vision. Par conséquent, nous cherchons à tirer parti des appareils mobiles et à explorer la possibilité de les utiliser pour traiter des calculs distribués. Une question qui en découle est donc *dans quelle mesure les dispositifs mobiles fournissent des ressources de calcul.*

Cette thèse se concentre sur une approche non-conventionnelle; pour la commodité du lecteur, nous la représentons dans la figure A.2. Elle consiste à décharger les tâches de calcul des serveurs MEC vers les appareils mobiles, effectuant ainsi le calcul *au-delà de la bordure.* De cette façon, les dispositifs mobiles servent de stations de travail mobiles potentielles pour aider à soulager la charge de l'infrastructure du réseau. Ils peuvent également améliorer la disponibilité des serveurs MEC. Ainsi, ils peuvent augmenter la performance globale des MEC. Dans cette thèse, nous abordons donc la mesure des ressources disponibles au-delà de la bordure, permettant l'extension des capacités MEC et le soulagement global de l'infrastructure réseau. Une telle approche est d'autant plus prometteuse que la pléthore d'appareils mobiles présents au-delà de la frontière. Ainsi, l'utilisation de ressources inexploitées par l'exécution de tâches semble prometteuse. Cependant, de nombreux défis doivent encore être relevés pour confirmer sa viabilité.

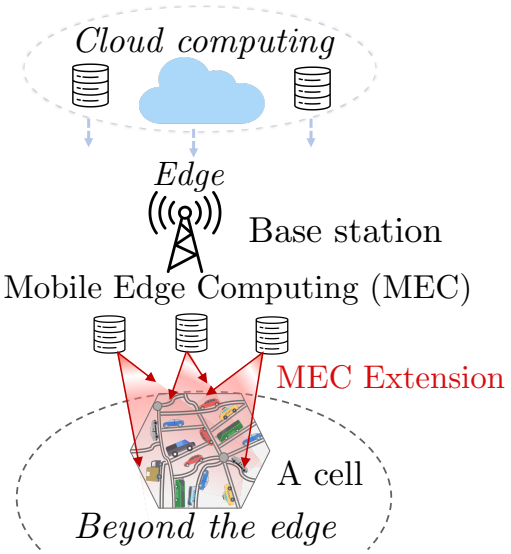


Figure A.2: Extension du MEC au-delà du bordure.

A.3 Challenges

Il est essentiel que l'orchestrateur MEC ait une vision instantanée et précise de ce qui se déroule au-delà de la bordure. Il doit être conscient de la dynamique de son environnement ou des ressources potentiellement disponibles à tous les niveaux. Cette prise de conscience est la clé pour améliorer sa gestion et l'exécution de ses tâches. Par exemple, un défi important pour décharger les tâches du MEC vers les appareils mobiles réside dans la complexité inhérente à la mobilité. En effet, les utilisateurs se déplacent avec leurs appareils lorsqu'ils communiquent au travail, dans les parcs, dans les centres commerciaux. Par conséquent, la mobilité affecte les ressources potentielles disponibles sur les appareils mobiles. De plus, pour que le MEC puisse s'appuyer sur les terminaux pour exécuter des tâches de calculs informatiques, il est fondamental que l'orchestrateur du MEC ait une idée de l'état des appareils mobiles à un moment donné. Par exemple, il faut pouvoir estimer si les dispositifs mobiles disposent de suffisamment de ressources de calcul, de stockage et de batterie, comme le montre la figureA.3. De plus, il est évident que sans ces conditions, un MEC est incapable de décharger une tâche. En outre, supposons que le MEC initialise la connexion avec un dispositif mobile aléatoire pour envoyer une tâche. Dans ce cas, il y a une forte probabilité de la perdre. C'est pourquoi l'identification des caractéristiques de mobilité est primordiale pour tirer parti des dispositifs mobiles. *Ainsi, au cours de cette thèse, nous nous concentrons principalement sur l'analyse des caractéristiques de mobilité des appareils mobiles.*

Cette thèse présente une nouvelle idée avant-gardiste qui consiste à exploiter les dis-

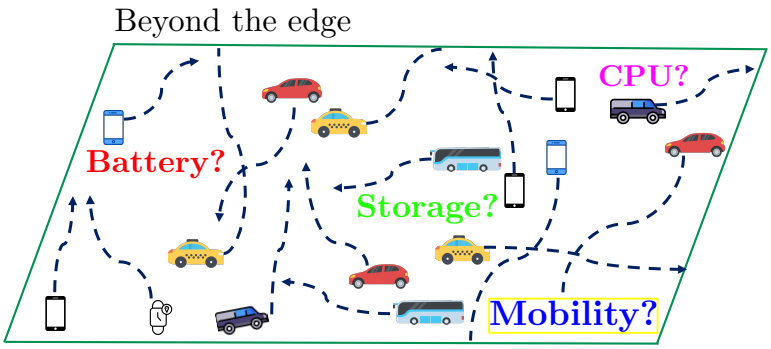


Figure A.3: Les défis à relever pour exploiter les appareils mobiles au-delà de la bordure.

positifs mobiles pour soulager l'infrastructure du réseau. De nombreuses recherches se sont concentré sur le délestage des dispositifs vers l'infrastructure réseau (c'est-à-dire le cloud, l'edge computing). Cependant, la première étape à notre étude est de savoir si la caractérisation de l'au-delà de la bordure a déjà été faite. Par exemple, existe-t-il une étude illustrant le potentiel des appareils mobiles au-delà de la périphérie ? De plus, comme nous l'avons mentionné précédemment, étant donné leur présence significative, comment pourraient-ils servir à étendre les capacités du MEC ? Ainsi, le MEC doit comprendre l'environnement des dispositifs mobiles avant d'en tirer parti. Par la suite, le MEC doit également estimer le volume des ressources disponibles qu'il peut utiliser à partir des dispositifs mobiles. De cette réflexion, les défis qui en découlent sont :

Challenge 1: À quelle fréquence et pendant combien de temps, les appareils mobiles restent-ils co-localisés, de sorte qu'ils puissent étendre, collectivement, les capacités du MEC ?

Cette question est cruciale, car elle nous permettra d'affirmer si nous pouvons utiliser les dispositifs mobiles. De plus, elle permettra d'éviter les échecs d'éxécution lorsque le MEC est sur le point d'attribuer des tâches de calculs informatiques aux dispositifs mobiles. Le cœur de cette question est la disponibilité temporelle des ressources au-delà du bord.

Challenge 2: Combien de tâches assignées par le MEC les appareils mobiles peuvent-ils entreprendre ?

Cette question donne un aperçu de la valeur des dispositifs mobiles pour l'extension du MEC. Nous portons donc notre attention sur l'allocation potentielle des tâches. Nous mettons l'accent sur l'aspect spatial en quantifiant les ressources qu'une cellule peut fournir pour effectuer une tâche MEC.

Nous évaluons l'utilisation des dispositifs mobiles en utilisant des jeux de données véhiculaires et piétons. Dans le cas des véhicules, nous considérons deux jeux de données : (i) 320 taxis à Rome, Italie [11] et (ii) 12 000 bus à Rio de Janeiro, Brésil [12]. Dans le cas des piétons, nous avons collecté un ensemble de données du réseau central d'un grand opérateur mobile européen. Il décrit le trafic mobile généré par tous les abonnés de Paris [13]. Ces ensembles de données sont intrinsèquement différents et nous permettent de mieux comprendre les possibilités offertes par les appareils mobiles en termes de déchargement potentiel.

Dans ce contexte, les contributions de cette thèse sont :

A.3.1 Contribution 1 : L'exploitation des dispositifs mobiles dans un scénario véhiculaire

.

Nous étudions la capacité de déchargement de calcul qu'un ensemble de véhicules dans une cellule peut offrir au MEC. Nous appelons la taille de cet ensemble la capacité de déchargement "potentielle" du cloud véhiculaire. Les véhicules étant très mobiles, il est fondamental d'étudier la dynamique de la capacité de déchargement potentielle. Par conséquent, lorsque le MEC est sur le point d'attribuer des tâches de calculs informatiques aux véhicules, il doit d'abord estimer la durée de vie des véhicules composant le cloud véhiculaire. En bref, la question que nous considérons est la suivante : "étant donné l'ensemble des véhicules dans une cellule à un instant t, quelle est la durée prévue de cet ensemble avant que sa taille ne diminue à un pourcentage donné par rapport à sa taille initiale ?". Nous appelons cette durée la *persistance* de la capacité de déchargement potentielle.

Une façon simple d'estimer la persistance est de se baser sur le temps de séjour des véhicules dans la cellule. Le temps de séjour est un élément crucial dans la caractérisation de la mobilité des utilisateurs dans les réseaux cellulaires [25, 26]. Cependant, dans la littérature, seules quelques études étudient le temps de séjour dans le contexte des véhicules [27]. En général, la mobilité des véhicules et les modèles de trafic sont déduits des connexions des téléphones mobiles, indiquant les modèles de déplacement [27]. D'autres techniques, basées sur des détecteurs de véhicules et des sondes GPS, mesurent

le temps de résidence des cellules [28]. Dans cette étude, contrairement aux temps de séjour qui caractérisent le comportement des véhicules sur une base individuelle, nous nous intéressons au comportement d'un ensemble de véhicules.

Nous évaluons la persistance de la capacité potentielle de déchargement directement à partir d'un ensemble de nœuds en utilisant des ensembles de données sur la mobilité des véhicules. Tout d'abord, nous introduisons un seuil de réduction α pour caractériser le point où nous supposons qu'un ensemble s'est dissous. Ensuite, nous évaluons la persistance des nœuds entre leur formation et leur dissolution – soit après une courte période (faible persistance), soit après une longue période (forte persistance). Nous évaluons également l'impact de la taille des cellules en la faisant varier. Enfin, nous soulignons les principales différences entre les ensembles de données et leur impact sur la capacité potentielle de déchargement des nuages véhiculaires. Le principal résultat de notre analyse est la caractérisation des périodes de temps qui doivent être réservées aux tâches légères et des périodes qui sont propices au déchargement des tâches qui nécessitent un temps de calcul plus long, grâce à une capacité de déchargement potentielle plus stable.

A.3.2 Contribution 2 : L'exploitation des dispositifs mobiles dans un scénario piéton

Nous abordons spécifiquement l'impact de la mobilité des dispositifs sur le succès de l'exécution des tâches dans un scénario piéton. Nous caractérisons une tâche comme une *durée*, c'est-à-dire le temps qu'il faut à un nœud mobile pour achever la tâche. Nous considérons en outre qu'un dispositif mobile doit achever la tâche dans un délai donné, que nous appelons *délai d'achèvement*. Pour évaluer le succès de l'exécution de la tâche, nous comparons la durée de la tâche avec le temps de présence du nœud dans la cellule. Cela nous permet d'obtenir le *nombre potentiel* de nœuds qui seraient capables d'exécuter la tâche. Nos analyses permettent d'évaluer l'impact de la mobilité des nœuds sur le potentiel de déchargement des tâches d'une cellule. D'une part, l'impact du délai d'achèvement montre que plus le délai d'achèvement est important, plus les possibilités d'exécuter une tâche sont nombreuses. D'autre part, il est inutile d'augmenter continuellement le délai d'achèvement, car les gains deviennent négligeables après un certain point. Ainsi, en fonction de la vitesse à laquelle le MEC doit exécuter ses tâches, il peut identifier le nombre maximum de tâches que les nœuds mobiles peuvent exécuter. Nos autres résultats indiquent que même les nœuds qui entrent et sortent de façon répétée d'une cellule peuvent contribuer au déchargement des tâches du MEC.

En somme, les deux questions sont étroitement liées ; les aspects temporels et spatiaux

vont de pair dans cette étude. La différence significative entre les deux travaux réside dans le type de mobilité et l'angle d'approche. En outre, l'un se concentre sur l'analyse de l'évolution d'un groupe de dispositifs mobiles et l'autre sur l'analyse du dispositif mobile individuel. Enfin, des seuils sont utilisés dans les deux études pour mieux comprendre l'impact de la mobilité. L'un s'assure qu'il y a suffisamment de dispositifs mobiles présents, et l'autre sert à récupérer plus de ressources. Notre façon de considérer ce sujet est résumée dans la figure A.4.

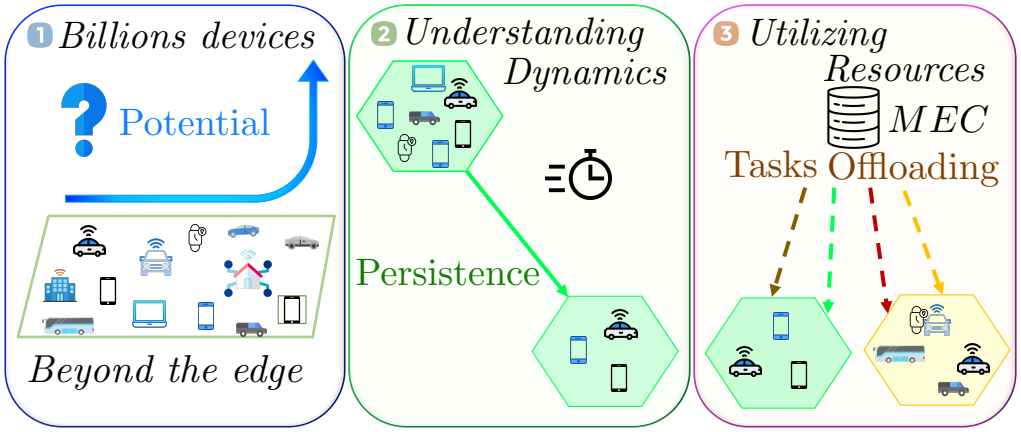


Figure A.4: Notre cheminement vers la réflexion sur cette thématique.

A.4 Conclusion

L'omniprésence des appareils mobiles et le désir d'améliorer l'expérience de l'utilisateur donnent lieu à des applications sophistiquées, dont beaucoup nécessitent une puissance de calcul importante sur les appareils des utilisateurs. La réalisation de ces applications nécessite des calculs intensifs et des contraintes de délais courts. L'une des approches techniques qui suscite une attention considérable pour améliorer l'efficacité du cloud computing est le MEC. Néanmoins, le MEC offre des ressources beaucoup plus limitées que le cloud et, dans certaines situations, il ne peut pas évoluer avec le nombre de tâches. Une idée séduisante consiste à étendre la capacité du MEC en exploitant les ressources disponibles sur les appareils des utilisateurs finaux.

Dans cette thèse, d'une part, nous avons exposé la possibilité d'utiliser des dispositifs mobiles pour multiplier les possibilités de calcul. En effet, à travers la formation de groupe de calcul disponible, leur persistance spatio-temporelle et en faisant varier la taille des cellules, nous avons affirmé la possibilité de décharger les tâches du MEC. Ainsi, sur la

base de nos jeux de données, il est possible d'étendre les capacités du MEC au-delà du bord. De plus, nos résultats montrent que le temps de persistance des groupes suit un modèle qui dépend de la densité du trafic véhiculaire et de l'heure de la journée. L'analyse des tendances de persistance nous amène à penser que les zones périurbaines, les zones de transit, sont adaptées à un déchargement court et inversement pour les zones urbaines. Par conséquent, nous suggérons de dissocier les régions à forte et à faible affluence pour obtenir une cartographie plus précise. Deuxièmement, nous quantifions combien de ressources peuvent être soustraites en envoyant des tâches de différentes tailles et en faisant varier le temps d'achèvement autorisé par la MEC. Nos résultats montrent que chaque appareil mobile contribue comme un poste de travail mobile permettant au MEC de décharger ses tâches. Par conséquent, en fonction du nombre de tâches que le MEC doit décharger, il peut sélectionner la cellule la plus appropriée. En conclusion, même sans le temps d'achèvement, nous disposons de plusieurs possibilités de déchargement pour la MEC. La faisabilité de l'exécution des tâches augmente de manière significative en utilisant l'achèvement. La tolérance élevée du MEC à l'égard du délai d'achèvement permet de se rapprocher de la limite maximale de calcul qu'un dispositif mobile peut fournir.

A.5 Perspectives

Dans ce manuscrit, nous avons présenté nos contributions pour le domaine de recherche MEC. Nous prévoyons prochainement d'étendre nos travaux actuels, car, pour l'instant, notre travail n'en est qu'au stade de l'observation-déduction. Bien que cette thèse propose un angle de recherche non-conventionnel, nous pensons qu'il peut être étendu à des directions plus complètes afin de profiter pleinement des capacités des appareils mobiles. Dans ce qui suit, nous énumérons quelques directions qui peuvent être étudiées dans des travaux futurs :

- **Prédiction du temps de persistance d'un groupe de nœuds.** Nous voulons proposer une métrique/modèle pour prédire le temps de persistance d'un groupe de nœuds. Le profilage des capacités offertes par les appareils mobiles peut être une étape importante pour le processus de déchargement des tâches. De plus, nous prévoyons également d'étudier la persistance dans un plus grand nombre de villes, dans des cellules de tailles différentes et sur des périodes plus longues afin d'améliorer l'analyse de l'évolution de la persistance.

- **Choisir une cellule appropriée dans laquelle décharger une tâche.** L'utilisation efficace des dispositifs mobiles est essentielle pour éviter les pertes de temps. C'est

pourquoi le choix des emplacements appropriés et des nœuds adéquats pour décharger une tâche est primordial pour l'extension MEC. Par conséquent, nous considérons une métrique qui combine le temps de persistance et le délai d'achèvement avec les propriétés de la tâche pour cibler un meilleur déchargement des tâches.

- **Estimation des capacités de stockage, de calcul et de batterie.** Il ne suffit pas d'avoir un groupe de dispositifs mobiles co-localisés et de connaître leur dynamique pour avoir une estimation complète des possibilités de déchargement. Nous devons prêter attention à d'autres paramètres pour évaluer et affiner une métrique potentielle de déchargement. C'est pourquoi nous nous concentrerons sur la capacité de la batterie des appareils et sur leur capacité de calcul et de stockage dans nos travaux futurs. Par exemple, les appareils mobiles peuvent rester ensemble pendant une longue période. Cependant, si leurs batteries sont insuffisantes pour exécuter une tâche informatique, le groupe est inutile pour le déchargement d'une tâche.

- **Modélisation générale des ressources disponibles au-delà de la périphérie.** Notre objectif est de proposer une métrique holistique pour donner une compréhension plus représentative des ressources au-delà du bord. Cette partie du travail permettra de rassembler les différentes parties de notre travail. En effet, la co-localisation/persistance et l'estimation de la capacité de stockage, de calcul et de batterie afin d'obtenir une cartographie complète des ressources au-delà du edge computing.

- **Système de tarification ou de récompense pour les utilisateurs.** Les appareils mobiles appartiennent à leur propriétaire. Ainsi, un modèle de tarification pour l'utilisation des ressources joue un rôle crucial. L'utilisateur doit avoir une raison de laisser son dispositif mobile exploiter pour le calcul des tâches MEC.

- **Problèmes de sécurité.** Le déchargement des calculs signifie que la MEC envoie le code source et les données de sa tâche à un tiers, ce qui peut entraîner un problème de sécurité important. Comment s'assurer qu'aucune copie des données du MEC n'est effectuée ? En outre, le serveur de périphérie doit également faire confiance au code source de l'utilisateur. Par conséquent, le serveur périphérique et l'utilisateur doivent tous deux établir un échange de confiance. Enfin, comment s'assurer que la MEC ne surcharge pas les appareils mobiles des utilisateurs ? Ces questions sont très difficiles dans le contexte du délestage informatique. Par conséquent, il est impératif de considérer la sécurité comme une question essentielle pour l'utilisation large des appareils mobiles.

Bibliography

[1] Cisco. Cisco annual internet report. `https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html`, 2018–2023.

[2] P. Mach and Z. Becvar. Mobile edge computing: A survey on architecture and computation offloading. *IEEE Communications Surveys Tutorials*, 19(3):1628–1656, 2017.

[3] K. Sasaki, N. Suzuki, S. Makido, and A. Nakao. Vehicle control system coordinated between cloud and mobile edge computing. In *2016 55th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, pages 1122–1127, Sep. 2016.

[4] T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili. Collaborative mobile edge computing in 5g networks: New paradigms, scenarios, and challenges. *IEEE Communications Magazine*, 55(4):54–61, 2017.

[5] J. Zhang, W. Xia, F. Yan, and L. Shen. Joint computation offloading and resource allocation optimization in heterogeneous networks with mobile edge computing. *IEEE Access*, 6:19324–19337, 2018.

[6] Pasika Ranaweera, Anca Delia Jurcut, and Madhusanka Liyanage. Survey on multi-access edge computing security and privacy. *IEEE Communications Surveys Tutorials*, 23(2):1078–1124, 2021.

[7] Wen Zhang, Longzhuang Li, Ning Zhang, Tao Han, and Shangguang Wang. Air-ground integrated mobile edge networks: A survey. *IEEE Access*, 8:125998–126018, 2020.

[8] Aaron Carroll and Gernot Heiser. An analysis of power consumption in a smartphone. In *Proceedings of the 2010 USENIX Conference on USENIX Annual Technical Conference*, USA, 2010. USENIX Association.

[9] J. Haj-Yahya, Y. Sazeides, M. Alser, E. Rotem, and O. Mutlu. Techniques for reducing the connected-standby energy consumption of mobile devices. In *2020 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pages 623–636, 2020.

[10] Diala Naboulsi, Marco Fiore, Stephane Ribot, and Razvan Stanica. Large-scale mobile traffic analysis: a survey. *IEEE Communications Surveys & Tutorials*, 18(1):124–161, 2015.

[11] Lorenzo Bracciale, Marco Bonola, Pierpaolo Loreti, Giuseppe Bianchi, Raul Amici, and Antonello Rabuffi. CRAWDAD dataset roma/taxi (v. 2014-07-17). Downloaded from `https://crawdad.org/roma/taxi/20140717`, July 2014.

[12] Daniel Dias and Luís Henrique Maciel Kosmalski Costa. CRAWDAD dataset coppe-ufrj/riobuses (v. 2018-03-19). Downloaded from `https://crawdad.org/coppe-ufrj/RioBuses/20180319`, March 2018.

[13] The Cancan project. Content and context based adaptation in mobile networks.

[14] K. Zhang, Y. Mao, S. Leng, Y. He, and Y. ZHANG. Mobile-edge computing for vehicular networks: A promising network paradigm with predictive off-loading. *IEEE Vehicular Technology Magazine*, 12(2):36–44, June 2017.

[15] X. Chen, L. Jiao, W. Li, and X. Fu. Efficient multi-user computation offloading for mobile-edge cloud computing. *IEEE/ACM Transactions on Networking*, 24(5):2795–2808, October 2016.

[16] Guanhua Qiao, Supeng Leng, Ke Zhang, and Yejun He. Collaborative task offloading in vehicular edge multi-access networks. *IEEE Communications Magazine*, 56(8):48–54, 2018.

[17] Yun Zhao, Sheng Zhou, Tianchu Zhao, and Zhisheng Niu. Energy-efficient task offloading for multiuser mobile cloud computing. In *2015 IEEE/CIC International Conference on Communications in China (ICCC)*, pages 1–5. IEEE, 2015.

[18] Rui Zhang, Libing Wu, Shuqin Cao, Xinrong Hu, Shan Xue, Dan Wu, and Qingan Li. Task offloading with task classification and offloading nodes selection for mec-enabled iov. *ACM Trans. Internet Technol.*, 22(2), oct 2021.

[19] Zhong Wang, Guangtao Xue, Shiyou Qian, and Minglu Li. Efficient resources allocation and computation offloading model for ap-based edge cloud. In *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, pages 1–6, 2020.

[20] Weiwen Zhang, Yonggang Wen, K. Guan, Dan Kilper, Haiyun Luo, and Dapeng Wu. Energy-optimal mobile cloud computing under stochastic wireless channel. *Wireless Communications, IEEE Transactions on*, 12:4569–4581, 09 2013.

[21] Thinh Quang Dinh, Jianhua Tang, Quang Duy La, and Tony Q. S. Quek. Offloading in mobile edge computing: Task allocation and computational frequency scaling. *IEEE Transactions on Communications*, 65(8):3571–3584, 2017.

[22] Hong Xing, Liang Liu, Jie Xu, and Arumugam Nallanathan. Joint task assignment and wireless resource allocation for cooperative mobile-edge computing. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–6, 2018.

[23] Haneul Ko and Sangheon Pack. Distributed device-to-device offloading system: Design and performance optimization. *IEEE Transactions on Mobile Computing*, 20(10):2949–2960, 2021.

[24] T. Penner, A. Johnson, B. Van Slyke, M. Guirguis, and Q. Gu. Transient clouds: Assignment and collaborative execution of tasks on mobile devices. In *2014 IEEE Global Communications Conference*, pages 2801–2806, 2014.

[25] X. Lin, R. K. Ganti, P. J. Fleming, and J. G. Andrews. Towards understanding the fundamentals of mobility in cellular networks. *IEEE Transactions on Wireless Communications*, 12(4):1686–1698, April 2013.

[26] Paul Baumann, Wilhelm Kleiminger, and Silvia Santini. How long are you staying? predicting residence time from human mobility traces. In *Proceedings of the 19th Annual International Conference on Mobile Computing Networking*, page 231–234, New York, NY, USA, 2013. Association for Computing Machinery.

[27] R. A. Becker, R. Caceres, K. Hanson, J. M. Loh, S. Urbanek, A. Varshavsky, and C. Volinsky. A tale of one city: Using cellular network data for urban planning. *IEEE Pervasive Computing*, 10(4):18–26, April 2011.

[28] Juan C. Herrera, Daniel B. Work, Ryan Herring, Xuegang (Jeff) Ban, Quinn Jacobson, and Alexandre M. Bayen. Evaluation of traffic data obtained via gps-enabled mobile phones: The mobile century field experiment. *Transportation Research Part C: Emerging Technologies*, (4):568 – 583, 2010.

[29] Ericsson. Ericsson mobility report. `https://www.ericsson.com/assets/local/reports-papers/mobility-report/documents/2020/november-2020-ericsson-mobility-report.pdf`, Nov 2020.

[30] Mohsen Sharifi, Somayeh Kafaie, and Omid Kashefi. A survey and taxonomy of cyber foraging of mobile devices. *IEEE Communications Surveys & Tutorials*, 14(4):1232–1243, 2011.

[31] Khadija Akherfi, Micheal Gerndt, and Hamid Harroud. Mobile cloud computing for computation offloading: Issues and challenges. *Applied computing and informatics*, 14(1):1–16, 2018.

[32] Songtao Guo, Bin Xiao, Yuanyuan Yang, and Yang Yang. Energy-efficient dynamic offloading and resource scheduling in mobile cloud computing. In *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9. IEEE, 2016.

[33] Hua Peng, Wu-Shao Wen, Ming-Lang Tseng, and Ling-Ling Li. Joint optimization method for task scheduling time and energy consumption in mobile cloud computing environment. *Applied Soft Computing*, 80:534–545, 2019.

[34] Sokol Kosta, Andrius Aucinas, Pan Hui, Richard Mortier, and Xinwen Zhang. Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading. In *2012 Proceedings IEEE Infocom*, pages 945–953. IEEE, 2012.

[35] Helber Wagner da Silva, Felipe Rocha Barbalho, and Augusto Venâncio Neto. Cross-layer multiuser session control for optimized communications on sdn-based cloud platforms. *Future Generation Computer Systems*, 92:1116–1130, 2019.

[36] M. V. Barbera, S. Kosta, A. Mei, and J. Stefa. To offload or not to offload? the bandwidth and energy costs of mobile cloud computing. In *2013 Proceedings IEEE INFOCOM*, pages 1285–1293, 2013.

[37] Y. Liu, M. J. Lee, and Y. Zheng. Adaptive multi-resource allocation for cloudlet-based mobile cloud computing system. *IEEE Transactions on Mobile Computing*, 15(10):2398–2410, 2016.

[38] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie. Mobile edge computing: A survey. *IEEE Internet of Things Journal*, 5(1):450–465, 2018.

[39] Yun Chao Hu, Milan Patel, Dario Sabella, Nurit Sprecher, and Valerie Young. Mobile edge computing—a key technology towards 5g. 2015.

[40] Slaundefinedana Jošilo and György Dán. Joint allocation of computing and wireless resources to autonomous devices in mobile edge computing. In *Proceedings of the 2018 Workshop on Mobile Edge Communications*, MECOMM'18, page 13–18, New York, NY, USA, 2018. Association for Computing Machinery.

[41] Xiantao Jiang, F. Richard Yu, Tian Song, and Victor C. M. Leung. A survey on multi-access edge computing applied to video streaming: Some research issues and challenges. *IEEE Communications Surveys Tutorials*, 23(2):871–903, 2021.

[42] Chang Shu, Zhiwei Zhao, Geyong Min, and Shuowei Chen. Mobile edge aided data dissemination for wireless healthcare systems. *IEEE Transactions on Computational Social Systems*, 6(5):898–906, 2019.

[43] Mahadev Satyanarayanan. A brief history of cloud offload: A personal journey from odyssey through cyber foraging to cloudlets. *GetMobile: Mobile Computing and Communications*, 18(4):19–23, 2015.

[44] Ranesh Kumar Naha, Saurabh Garg, Dimitrios Georgakopoulos, Prem Prakash Jayaraman, Longxiang Gao, Yong Xiang, and Rajiv Ranjan. Fog computing: Survey of trends, architectures, requirements, and research directions. *IEEE Access*, 6:47980–48009, 2018.

[45] Pooyan Habibi, Mohammad Farhoudi, Sepehr Kazemian, Siavash Khorsandi, and Alberto Leon-Garcia. Fog computing: A comprehensive architectural survey. *IEEE Access*, 8:69105–69133, 2020.

[46] Jinke Ren, Guanding Yu, Yunlong Cai, and Yinghui He. Latency optimization for resource allocation in mobile-edge computation offloading. *IEEE Transactions on Wireless Communications*, 17(8):5506–5519, 2018.

[47] Juan Liu, Yuyi Mao, Jun Zhang, and Khaled Ben Letaief. Delay-optimal computation task scheduling for mobile-edge computing systems. *CoRR*, abs/1604.07525, 2016.

[48] Jeongho Kwak, Yeongjin Kim, Joohyun Lee, and S. Chong. Dream: Dynamic resource and task allocation for energy minimization in mobile cloud systems. *IEEE Journal on Selected Areas in Communications*, 33:2510–2523, 2015.

[49] Lixing Chen, Sheng Zhou, and Jie Xu. Computation peer offloading for energy-constrained mobile edge computing in small-cell networks. *IEEE/ACM Transactions on Networking*, 26(4):1619–1632, 2018.

[50] Shuchen Zhou and Waqas Jadoon. The partial computation offloading strategy based on game theory for multi-user in mobile edge computing environment. *Comput. Netw.*, 178(C), sep 2020.

[51] Wael Labidi, Mireille Sarkiss, and Mohamed Amine Kamoun. Energy-optimal resource scheduling and computation offloading in small cell networks. *2015 22nd International Conference on Telecommunications (ICT)*, pages 313–318, 2015.

[52] L. Tong, Y. Li, and W. Gao. A hierarchical edge cloud architecture for mobile computing. In *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9, 2016.

[53] Yuyi Mao, Jun Zhang, and Khaled B. Letaief. Dynamic computation offloading for mobile-edge computing with energy harvesting devices. *IEEE Journal on Selected Areas in Communications*, 34(12):3590–3605, 2016.

[54] Sennur Ulukus, Aylin Yener, Elza Erkip, Osvaldo Simeone, Michele Zorzi, Pulkit Grover, and Kaibin Huang. Energy harvesting wireless communications: A review of recent advances. *IEEE Journal on Selected Areas in Communications*, 33(3):360–381, 2015.

[55] Yeongjin Kim, Jeongho Kwak, and Song Chong. Dual-side optimization for cost-delay tradeoff in mobile edge computing. *IEEE Transactions on Vehicular Technology*, 67(2):1765–1781, 2017.

[56] H. R. Arkian, R. E. Atani, and A. Pourkhalili. A cluster-based vehicular cloud architecture with learning-based resource management. In *2014 IEEE 6th International Conference on Cloud Computing Technology and Science*, pages 162–167, 2014.

[57] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen. Vehicular fog computing: A viewpoint of vehicles as the infrastructures. *IEEE Transactions on Vehicular Technology*, 65(6):3860–3873, June 2016.

[58] Q. Liu, Z. Su, and Y. Hui. Computation offloading scheme to improve qoe in vehicular networks with mobile edge computing. In *2018 10th International Conference on Wireless Communications and Signal Processing (WCSP)*, pages 1–5, Oct 2018.

[59] J. Feng, Z. Liu, C. Wu, and Y. Ji. Ave: Autonomous vehicular edge computing framework with aco-based scheduling. *IEEE Transactions on Vehicular Technology*, 66(12):10660–10675, Dec 2017.

[60] K. Zhang, Y. Mao, S. Leng, S. Maharjan, and Y. Zhang. Optimal delay constrained offloading for vehicular edge computing networks. In *2017 IEEE International Conference on Communications (ICC)*, pages 1–6, May 2017.

[61] X. Sun and N. Ansari. Edgeiot: Mobile edge computing for the internet of things. *IEEE Communications Magazine*, 54(12):22–29, 2016.

[62] Lei Liu, Chen Chen, Qingqi Pei, Sabita Maharjan, and Yan Zhang. Vehicular edge computing and networking: A survey. 07 2019.

[63] Y. Guo, Q. Yang, F. R. Yu, and V. C. M. Leung. Cache-enabled adaptive video streaming over vehicular networks: A dynamic approach. *IEEE Transactions on Vehicular Technology*, 67(6):5445–5459, June 2018.

[64] J. Cui, L. Wei, J. Zhang, Y. Xu, and H. Zhong. An efficient message-authentication scheme based on edge computing for vehicular ad hoc networks. *IEEE Transactions on Intelligent Transportation Systems*, 20(5):1621–1632, May 2019.