A survey on mobility-induced service migration in the fog, edge and related computing paradigms

ZEINEB REJIBA, Advanced Network Architectures Lab (CRAAX), Universitat Politècnica de Catalunya (UPC), Spain
XAVIER MASIP-BRUIN, Advanced Network Architectures Lab (CRAAX), Universitat Politècnica de Catalunya (UPC), Spain
EVA MARÍN-TORDERA, Advanced Network Architectures Lab (CRAAX), Universitat Politècnica de Catalunya (UPC), Spain

With the advent of fog and edge computing paradigms, computation capabilities have been moved towards the edge of the network to support the requirements of highly demanding services. To ensure the quality of such services is still met in the event of users' mobility, migrating services across different computing nodes becomes essential. Several studies have emerged recently to address service migration in different edge-centric research areas, including fog computing, multi-access edge computing (MEC), cloudlets and vehicular clouds. Since existing surveys in this area either focus on VM migration in general or migration in a single research field (e.g. MEC), the objective of this survey is to bring together studies from different, yet related, edge-centric research fields, while capturing the different facets they addressed. More specifically, we examine the diversity characterizing the landscape of migration scenarios at the edge, we present an objective-driven taxonomy of the literature and we highlight contributions that rather focused on architectural design and implementation. Finally, we identify a list of gaps and research opportunities based on the observation of the current state of the literature. One such opportunity lies in joining efforts from both networking and computing research communities to facilitate future research in this area.

CCS Concepts: • **General and reference** → **Surveys and overviews**; • **Computer systems organization** → **Distributed architectures**; *Cloud computing*; • **Human-centered computing** → *Mobile computing*;

Additional Key Words and Phrases: fog computing, edge computing, vehicular clouds, service migration

ACM Reference Format:
Zeineb Rejiba, Xavier Masip-Bruin, and Eva Marín-Tordera. 2019. A survey on mobility-induced service migration in the fog, edge and related computing paradigms. *ACM Comput. Surv.* 1, 1, Article 1 (January 2019), 32 pages. https://doi.org/10.1145/3326540

1 INTRODUCTION

Fog computing and related edge-centric computing paradigms have been recently proposed to enhance the performance of an ever-increasing set of highly demanding applications, such as Internet of things (IoT) applications, augmented reality, virtual reality, gaming or smart surveillance systems, etc. To that end, a "one-hop" communication to the end user is set in order to meet the

---

Authors' addresses: Zeineb Rejiba, Advanced Network Architectures Lab (CRAAX), Universitat Politècnica de Catalunya (UPC), , Barcelona, Spain, zeinebr@ac.upc.edu; Xavier Masip-Bruin, Advanced Network Architectures Lab (CRAAX), Universitat Politècnica de Catalunya (UPC), Barcelona, Spain, xmasip@ac.upc.edu; Eva Marín-Tordera, Advanced Network Architectures Lab (CRAAX), Universitat Politècnica de Catalunya (UPC), , Barcelona, Spain, eva@ac.upc.edu.

0360-0300/2019/1-ART1 $15.00
https://doi.org/10.1145/3326540

expected strict latency requirements of these applications. Moreover, such decentralized near-edge computing facilities also contribute to alleviate the burden on the core network infrastructure, compared to the case where application-generated data was exclusively sent to remote cloud data centers. In short, fog/edge computing is intended to connect the cloud to the IoT domain, building a distributed set of nodes (referred to as Fog Nodes, see [59]) nearing computing and storage capacities to end users.

However, when considering the users' mobility along with the limited coverage of the fog nodes (FNs) serving them, the user-FN communication may need to go through multiple hops, which may severely affect the delivered quality of service (QoS), particularly for highly demanding services. In order to mitigate as much as possible the negative effects of such a QoS degradation, service execution must be dynamically migrated to a better placement, optimally, much closer to the new user location.

Such a need for migration at the edge can be observed in the advent of the "Follow Me" trend, where terms such as Follow Me Cloud[88, 89], Follow Me Edge[86] Follow Me Edge Cloud[3], Follow Me Fog[8], Move With Me[18], Companion Fog Computing[75], have recently emerged in the literature, further emphasizing the tight link between user mobility and service mobility.

Those works along with many others reiterate the need for little to no disruption of the running service during migration, along with the need to balance costs and benefits resulting from the migration. Certainly, this migration process, referred to as mobility-induced service migration, is highly challenging since it depends on several aspects including the user-FN distance (both the physical and the network distance), the size of the service to be migrated, the real available bandwidth and the actual load of the destination node, just to name a few.

While related topics such as virtual machine migration in cloud datacenters or mobile offloading decision-making have matured over the years, the topic of mobility-induced service migration at the edge is still under active research. With this in mind, in this survey, we provide a holistic view on the broad landscape of the contributions made so far in the literature and identify the gaps and future directions needed to drive the research in this field forward.

1.1 Related surveys

Recently, many survey papers have been published in the fog and edge computing areas. In a first categorization, we may differentiate between generic surveys (i.e., focusing on architectures, concepts, management approaches, etc.) and specific ones (i.e., dealing with security, privacy, big data management, etc.). In fact, many of the general-purpose surveys identify service migration as one of the key issues that need to be addressed in edge-centric environments. Table 1 summarizes this preliminary classification, including the most relevant surveys, also briefly introduced next.

We start with the Multi-Access Edge Computing (MEC) area, where authors in [90] identify MEC service mobility as one of the challenges that need to be addressed in order to ensure an efficient network and service orchestration in a MEC environment. [58] and [52] present computation migration as one of the resource management techniques in a MEC system. Both present a short overview of relevant contributions in this area. Similarly, in [93], migration is presented as one of the perspectives pertaining to resource allocation at the edge and as such, some relevant contributions in this area are reviewed. [7] discusses how Software-Defined Networking (SDN) can be beneficial in an edge computing scenario and identifies VM mobility as one of the areas that can significantly benefit from such a paradigm.

As for fog computing-related surveys, a number of migration-related papers have been reviewed in [62] in the context of offloading and load redistribution. Similarly, authors in [103] present another set of migration-related contributions in the context of Resource Management and Provisioning. Authors in [43] emphasize the need for location-awareness when performing VM/container

Table 1. Related surveys and studies

Research area	Study	Scope	
		Generic	"Migration at the Edge"-specific
Edge computing	[90]	✓	
	[58]	✓	
	[52]	✓	
	[93]	✓	
	[7]	✓	
	[99]		✓
	[76]		✓
Fog computing	[62]	✓	
	[103]	✓	
	[43]	✓	
	[74]	✓	
VM migration	[105]	✓	

migrations in fog/edge computing, while authors in [74] highlight some of the challenges that have to be addressed to support mobility in fog/edge environments.

Other than general purpose surveys, there exist a few surveys specifically dealing with migration at the edge. An example survey can be found in [99], yet with a primary focus on the MEC context. Another relevant study is conducted in [76] but it only studies the problem from a virtualization technology perspective.

With a focus rather targeted towards providing a thorough review on VM migration in general, authors in [105] also include a dedicated section on "user mobility-induced VM migration" in MEC. Authors divide the reviewed contributions into two broad categories: migration performance improvement and migration performance analysis. Even though their classification can be used as a basis for more detailed analyses, it does not capture the different facets and areas involved in the migration landscape and it only focuses on the VM technology in the MEC scenario.

Therefore, the observation of related surveys motivates us to combine and further develop the aforementioned efforts into a complete survey where we expose the different facets involved in the rich literature dealing with service migration in edge-centric environments.

1.2 Scope and contributions

As introduced earlier, this survey primarily focuses on mobility-induced service migration at the edge. Therefore, related contributions dealing with VM migration within or across data centers or those dealing with computation offloading from mobile devices to the cloud or to cloudlets lie out of the scope of this survey. For an overview of such contributions, the interested reader may refer to [2, 105] for the former, and [111] for the latter. In this work, we specifically contribute in the following directions:

- We identify all relevant edge-centric computing paradigms where service migration has been studied, regardless of the specific application scenario. These paradigms include cloudlets, fog computing, cloud-based vehicular networks as well as multi-access edge computing. The aim is to bring together into a single place the best of the advances made in these research areas with regards to service migration.
- We outline the high-level characteristics that can affect a given migration setup, therefore highlighting the most- and the least-commonly considered characteristics.

- We provide a detailed classification of the reviewed works based on their objectives, in addition to highlighting the works where the primary focus has been on architectural design and implementation.

The remainder of this paper is organized as follows. Section 2 outlines the different edge-related research areas where the topic of service migration has been studied. Section 3 presents a view on the diversity in the migration scenarios landscape. Section 4 reviews some basic terms and definitions generally encountered in the literature, thus setting the ground for the detailed analysis of the taxonomy proposed in Section 5. Section 6 presents the different architectures, platforms and implementations related to the migration at the edge. Section 7 lists the gaps identified in the reviewed literature, thus providing directions for future research. Finally, Section 8 concludes this survey.

2 A BACKGROUND REVIEW ON THE CONSIDERED RESEARCH AREAS

In this section, we outline the different edge-related research areas where the topic of mobility-induced service migration has been studied. Then, a unified view where the different considered areas can co-exist is presented.

2.1 Considered research areas

The topic of service migration was found to be relevant in several edge-related research areas. These include cloudlets, fog computing, cloud-based vehicular networks as well as multi-access edge computing. Since different terms have been used in the literature to refer to similar edge-related concepts, we briefly review the definitions of these research areas and we describe how service migration fits in each area.

2.1.1 Cloudlets. The first definition of cloudlets [81] goes back to 2009, where it has been introduced as “a trusted, resource-rich computer or cluster of computers that is well-connected to the Internet and is available for use by nearby mobile devices”. The term Micro Data Center (MDC) may alternatively be found in the literature to refer to a cloudlet. It was originally coined by Microsoft in [6].

According to [81], cloudlets and Wi-Fi access points can be combined within a single entity, where VM-based virtualization techniques are leveraged. Ultra-short responses can therefore be guaranteed for real time applications, thanks to the one-hop, high bandwidth connectivity to the cloudlet. However, given the inherent Wi-Fi range limitations, VMs belonging to users on the move may need to be migrated across cloudlets to maintain the benefits brought by the single hop connectivity links.

2.1.2 Fog computing. Fog Computing has been defined in [16] as a “highly virtualized platform that provides compute, storage, and networking services between end devices and traditional Cloud Computing Data Centers, typically, but not exclusively located at the edge of network”. The fog is usually characterized with low latency, predominance of wireless access, location awareness, geographical distribution as well as support for mobility and real time applications. Such characteristics strongly justify the need for efficiently migrating those applications from one fog node to the other following user mobility.

2.1.3 Cloud-based vehicular networks. As introduced in [104], cloud-based vehicular networks enhance conventional vehicular networks with cloud computing principles, therefore facilitating resource sharing among vehicles. The resulting architecture is hierarchical in nature, starting from a vehicular cloud comprised of nearby cooperative vehicles, a roadside cloud co-located with the Roadside Unit (RSU), up to the conventional cloud. [104] envision different migration scenarios in this context, including the migration from one vehicle to another vehicle in the same RSU area, or from one RSU-Cloud to another RSU-Cloud, or from an RSU-Cloud to a vehicle and vice-versa.

2.1.4 Multi-Access edge computing. The Mobile Edge Computing initiative[60] emerged in 2014 with an aim to provide “cloud-computing capabilities and an IT service environment at the edge of the network”. However, in 2017, the name was changed to Multi-Access Edge Computing to extend the scope to other radio access technologies. Alternative terms such as (mobile) edge clouds and mobile micro-clouds[97] may also be found in the literature to refer to MEC. Connected cars, augmented reality and video analytics constitute potential application areas for MEC. Migration needs to be supported in this context to ensure the quality required by those applications is met in the event of end users’ mobility.

2.2 Unified view

Although migration in the literature is usually addressed in one specific area of the aforementioned areas, we envision a scenario where all such areas co-exist and complement each other. More specifically, we consider that a fog node[1] , i.e. the host providing computation, storage and networking resources at the edge, can be either static or on the move. FNs can have various access technologies, including cellular, Wi-Fi or vehicular communications. Indeed, the radio access that they are leveraging would determine the scope of their area of operation. As a result, such a unified view creates several possible migration scenarios (marked with the orange arrows in Figure 1) that have been addressed in the literature in varying degrees of occurrence.

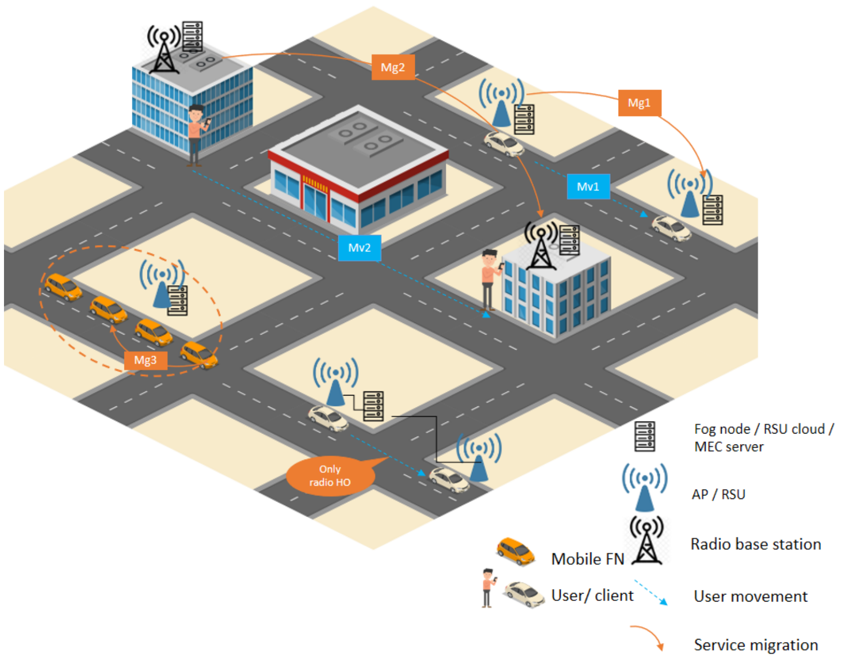

Fig. 1. Migration scenarios at the edge.

---

[1] We will adopt the term Fog Node (FN) for the rest of this paper, except in parts where other terms have been considered by the authors of the reviewed contributions.

3 A VIEW ON THE DIVERSITY IN THE MIGRATION SCENARIOS LANDSCAPE

In this section, a high-level classification showing the general characteristics of the migration contexts is presented with the aim of calling attention to the contexts that have been explored the least in the literature. A different classification of the proposed migration approaches rather focusing on their main objectives will be provided in Section 5.

The first characteristic that we examine is the migration element, i.e. “what is to be migrated?”. Different migration elements have been considered in the literature, each having a different granularity level. More specifically, a service[2] , which is the most commonly used term in the literature, has the highest granularity in most of the cases, whereas more fine-grained migration elements such as operators, application components or virtual objects, can be combined to create more complex services. In the latter case, they might need to be jointly migrated to meet their dependency requirements. Table 2 captures the different terms used in the literature to refer to the migration element, their definitions, as well as their corresponding examples.

Table 2. What is migrated? - Generic

Migration element	References	Definition	Example
Service	[89] [65] [96] [95] [47] [97] [98] [24] [94] [4] [107] [56] [106] [109] [55] [35] [72] [3] [32] [57] [1] [19] [66] [110] [54] [22]	Highest level asset describing the user’s objectives	Video service
Task	[100] [23] [112]	Part of an application[100][112]	Face recognition[23], video streaming[112]
Operator	[70] [44] [71]	Performs online processing of events, using a VM as an execution environment[70]	Car speed pattern detection, lane switch detection
Application (component)	[83] [82] [80]	Could be hosted in VMs or containers	Motion detection, face recognition
Job	[8] [9]	Delay-sensitive computation	-
Virtual object	[18]	Software component	Digital media server, edge storage
Cyber function	[48]	Provides data and control services to physical objects, feedback control as an example, virtualized using VMs or containers	Feedback control
Virtualized function	[13]	Created by applying an overlay on the VM base image	openCV
Avatar	[31][85]	Software clone that offers a service to the user wherever it moves, hosted within a VM in a cloudlet	-

The migration of the generic elements shown in Table 2 is often interchangeably used to refer to the migration of the virtual environment in which they run, such as VMs or containers. On the other hand, some works, listed in Table 3, primarily focus on the migration of the virtual environment itself. As it can be seen, VM migration has been extensively studied in the literature, however, increasing interest is being shown to more lightweight alternatives such as containers or processes in more recent works.

Apart from the diversity affecting the migration elements, a variety of considerations have been taken regarding the migration scenarios, which may in some cases be due to the characteristics of the corresponding research area. We particularly identify the five following considerations:

---

[2] We will adopt the term service in the rest of this paper, except in parts where other terms have been considered by the authors of the reviewed contributions.

Table 3. What is migrated? - Execution environment view

Execution environment	VM	Container	Process
Relevant literature	[40] [92] [46] [41] [102] [15] [53] [69] [33] [37] [104] [108] [101] [79] [11] [64] [84] [30] [21]	[78] [91] [75] [86] [55] [27] [38] [1]	[45] [19]

- (C1) Mobility: As mobility may affect both end users (U) and fog nodes (FN) (e.g. bus-based, vehicle-based, robot-based), we classify works according to whether they consider users' mobility only, fog node mobility only or both at the same time.
- (C2) FN-Radio Access mapping (applicable in the case of user mobility):
  - (1-1): One FN within each radio access area, it could be the same node providing radio access and fog node capabilities. In this case, both radio handover and service migration will take place, when the user moves across radio access areas (See Mg1 and Mg2 in Figure 1).
  - (N-1): Multiple FNs per radio access area. This is usually a characteristic of a dense deployment. When the user moves across areas, in addition to the radio handover, a selection of the FN to migrate to out of all the FNs in the area will take place.
  - (1-N): One FN is shared among multiple radio access coverage areas, in this case, as the user moves across these coverage areas, only the radio handover will occur, and thus migration is not needed (as depicted in the lower right part of Figure 1).
- (C3) Migration scope: This differentiates between the case where the migration has to be maintained within the same area (SA), this can be found in particular in a vehicular cloud scenario (see Mg3 in Figure 1), or among nearby areas (AA, see Mg1 and Mg2 in Figure 1).
- (C4) Migration timing: We distinguish migrations that start proactively (P), before the radio handover, while the user is still at the current area, and migrations which are performed reactively (R), after the radio handover.
- (C5) Migration elements to move at a given time: Single (S), e.g. a single self-contained service or multiple (M) components with dependencies between them. In the latter case, the components having a high interdependency rate need to be co-migrated together to the same locality as suggested in [18]. It is worth noting that there are some works that consider migrating multiple elements without explicit consideration of the aforementioned "co-migration" issue[3] .

A look at Table 4, where the considered literature has been classified according to the aforementioned criteria, allows us to draw the following observations:

- With regards to (C1), there are a few works considering the case of a mobile FN, and even fewer considering both users' and FNs' mobility.
- As for (C2), the migration within the same area is not commonly tackled, except in the case of a vehicular scenario.
- For (C3), authors usually consider a 1-1 mapping between the FN and the radio access technology it is associated to, especially in the case of MEC. However, with the advent of the network densification trend, more instances of the N-1 mapping may be observed in the future.
- Both proactive and reactive approaches have been almost equally studied in the literature (C4). Choosing one approach over the other depends on many considerations. For instance, while in [15], authors recommend the proactive approach to deal with delays that may be

---

[3] These are NOT marked with an asterisk in the corresponding column in Table 4.

Table 4. General aspects-Regarding mobility, User (U) Vs. Fog Node (FN); Migration boundaries: Same Area (SA) Vs. Among areas (AA); Timing: Proactive (P) Vs. Reactive (R); Elements to move at a given time: Single (S) Vs. Multiple (M)

Research area	Reference	Mobility		Migration scope		FN-RA mapping			Timing		Migration elements to move at a given time	
		U	FN	SA	AA	1-1	N-1	1-N	P	R	S	M
Cloudlets Micro data centers (MDCs)	[40] [31] [78]	✓			✓	✓			-	-	✓	
	[92] [89] [46] [65]	✓			✓	✓				✓	✓	
	[96]	✓			✓	✓			✓		✓	
	[100]	✓	✓	-	-	-	-	-	-	-		✓
	[23]	✓	✓	✓			✓		-	-	✓	
Fog computing	[102]	✓			✓		✓			✓	✓	
	[15]	✓			✓	✓			✓	✓	✓	
	[83]	✓			✓	✓			✓	✓		✓
	[53]	✓			✓	✓			✓		✓	
	[8],[9]	✓			✓	✓			✓			✓
	[95]	✓			✓	✓			-	-		✓(*)
	[47]	✓		-	-	-	-	-		✓	✓	
	[82]	✓			✓	✓				✓	✓	
	[91]	✓			✓	✓			-	-	✓	
	[33]	✓			✓		✓			✓	✓	
	[18]	✓			✓	✓				✓		✓(*)
	[112]	✓	✓	✓			✓		✓		✓	
	[14]	✓			✓	✓			✓	✓	-	-
	[37]	✓			✓	✓			✓		✓	
	[75]	✓		✓	✓		✓			✓	✓	
	[70][44][71]	✓			✓	✓			✓			✓
Cloud-based vehicular networks	[104],[108]	✓	✓	✓	✓		✓	✓		✓	✓	
	[101]	✓			✓	✓				✓	✓	
	[79][11] [64]		✓	✓			✓		✓		✓	
Multi-access edge computing Mobile Edge Computing (Mobile) Edge Clouds Mobile Micro Clouds (MMCs)	[97] [98] [24][94] [4][72] [3][84] [106][109] [30][66]	✓			✓	✓				✓	✓	
	[107][85] [35] [57] [56][38] [80]	✓			✓	✓			-	-	✓	
	[21] [86] [55] [54] [32][22]	✓			✓	✓			✓		✓	
	[13][1]	✓			✓	✓			✓	✓	✓	
	[19]	✓			✓	✓				✓		✓
	[110]	✓			✓	✓			-	-		✓
Others	[48]	✓			✓	✓				✓	✓	

experienced in the reactive approach if the amount of data to be transferred is high, authors in [13] consider the reactive approach as a backup mechanism to the cases of unexpected or unpredictable mobility. Indeed, in some contributions, the use of both approaches is envisioned.

- Finally, regarding (C5), there has not been much attention on the migration of inter-dependent elements, which may be needed in real-world scenarios to support complex services.

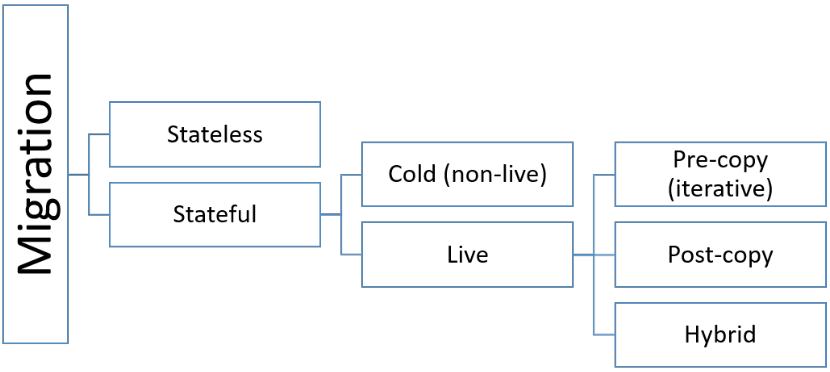

Fig. 2. Migration types.

4 COMMON TERMS AND CONCEPTS

Before going deeper in the analysis of the reviewed contributions, we briefly review some important terms and concepts usually found in the literature and that will be often used in Section 5. These terms are related to common migration types and costs, as well as the benchmarks usually used to evaluate migration strategies.

4.1 Migration types

As shown in Figure 2, migrations (of either VMs or containers) can be classified in two main categories: **stateless** and **stateful**. A stateless migration is associated to a stateless service that does not have a running state and as such, the migration process simply consists in re-deploying the service at the target node. On the other hand, a stateful migration has to ensure that the running state of the application is moved to the target node. Interactive applications such as gaming constitute some example applications that require stateful migration.

Stateful migration itself can be also divided into **cold** (alternatively **non-live**) and **live** migration. In cold migration, the VM is suspended and then transferred to the destination where it will be resumed. During that time, the service is unavailable at the current host. This period during which the service is unavailable is referred to as **downtime**. In the non-live migration, the service downtime is excessively long, thus negatively affecting the user experience. That is why, live migration is introduced to address this issue.

Live migration can be further divided to **pre-copy**, **post-copy** and **hybrid** migrations. To better understand the latter migration patterns, three basic phases of VM migration have been defined in [25] (also reported in [105]):

- A *push phase*, during which memory pages are iteratively pushed to the destination, while the VM is still running on the source. “Dirty” pages, i.e. the pages that have been modified since the last iteration, will be sent again in subsequent iterations.
- A *stop-and-copy phase*: The VM on the source is stopped. Remaining pages are then copied to the destination, where the VM will be restarted.

- A *pull phase*: As the VM resumes execution at the destination, page faults may occur when accessing pages which have not been copied yet. These pages will then be pulled from the source node during this phase.

The order in which these phases are executed determines the live migration type. In fact, pre-copy migration performs the Push phase then the Stop-and-copy phase, while the post-copy migration starts with the Stop-and-Copy and then executes the Pull phase. The hybrid approach combines phases from both pre- and post-copy approaches. Given the iterative copying phase in the pre-copy migration, a long **migration time**, i.e. the time between the migration initiation and the time when the VM/container is resumed at the destination, will be observed. This motivates the emergence of different works aiming to optimize the pre-copy process and that will be covered later in Section 5.2.1.

From a technical implementation perspective, most VM managers come with built-in support for live migration, such as Xen[10], KVM[49], etc. As for containers, the Checkpoint/Restore In Userspace (CRIU) tool [26] is commonly used in the literature to perform live container migrations.

4.2 Common costs definition

As will be shown later in Section 5.1.1, assessing migration-related costs is essential for an optimal migration decision making. In this context, the two most common costs considered in the literature are the migration and the transmission costs, as introduced in the Follow-Me-Cloud concept[87]. In fact, the **migration cost** is incurred when the service is moved from the previously-serving node to the one serving the new user location. As shown in [87], the service size, the service initiation/release cost and the bandwidth consumed to migrate the service are key elements that affect the migration cost. The **Transmission cost** is instead incurred when the service is not migrated and the user accesses the service in its originally-serving node via the backhaul network, instead of a one-hop link, therefore resulting in an increased latency.

4.3 Benchmarks

There exist common benchmarks that authors use to evaluate the performance of their proposed migration strategies.

These include **always migrate** (also known as **greedy** strategy or **least hop** strategy). This refers to the case where migration occurs each time a one-hop connection to a fog node is found. High migration costs might be experienced in this case, with potentially no significant improvement in the perceived quality, especially if the transferred size is large. Alternatively, the **no migration** strategy (i.e. **static** strategy) refers to the case where the service is not migrated but instead kept in its original placement and is accessed via backhaul links. In this case, no migration cost will be incurred at the expense of a potentially higher latency due to the backhaul communication. In addition, the **least loaded** strategy consists in migrating to the node with the lowest load, even if it is not one hop away from the user.

To evaluate the benefits of proactive migration approaches, authors usually compare to a **lazy** (also called **reactive** or **on-demand**) solution that performs migrations after the radio handover occurs. On the other hand, works proposing prediction mechanisms to enhance migration performance use an **oracle** benchmark that provides exact predictions. This allows obtaining an upper bound on the quality of the predictions. Finally, **myopic** approaches are used to refer to instantaneous cost optimizations, instead of considering future, long-term ones.

Other alternatives may also be found, however, we do not list them here since they are very specific to the considered problem.

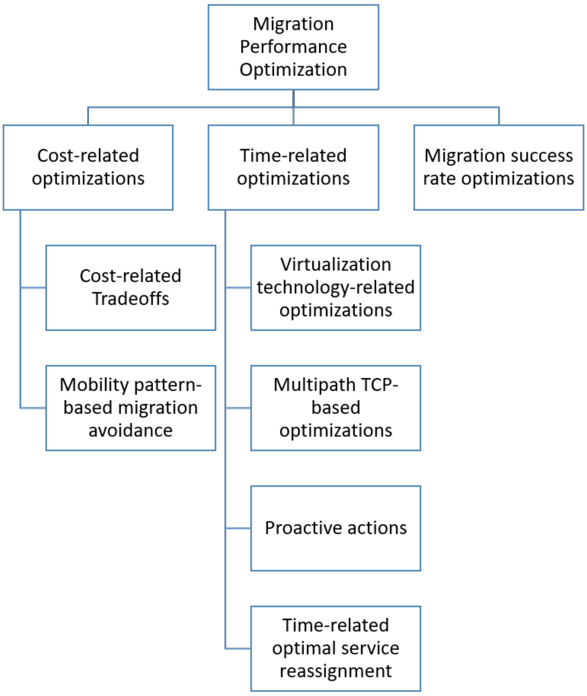

Fig. 3. Literature classification by objective

5 PROPOSED TAXONOMY

Figure 3 provides an overview of the proposed taxonomy for classifying the literature according to the addressed objective. As it can be seen, three main objectives have been identified. The first one is dealing with cost-related optimizations, which can be further divided into contributions dealing with cost-related tradeoffs and contributions proposing to avoid migrations by leveraging users' mobility patterns. The second objective is related to optimizations focusing on the time axis, either by considering the virtualization technology, the use of Multipath TCP, taking proactive actions or optimizing the service re-assignment. Finally, the third objective deals with migration success rate optimizations. Subsequent sections provide further details about this taxonomy. We note that certain contributions may fit under more than one category, in which case, they are assigned to the most relevant category.

5.1 Cost-related optimizations

In this section, we outline the different contributions that address the migration problem by analyzing its related costs and optimizing them. We distinguish contributions focusing on the cost-related tradeoffs and the ones which aim at avoiding costly migrations by leveraging information about the user mobility path.

5.1.1 Cost-related tradeoffs. Whenever a migration decision has to be taken, a tradeoff has to be made between the potential benefit that would result from migrating a service (e.g. QoS improvement) and the cost that may be incurred from doing so. Alternatively, the tradeoff may deal with the assessment of the impact of different types of costs on the performance. To address and model these tradeoffs, different approaches have been taken, including the use of Markov Decision

Processes (MDPs), different optimization techniques as well as relying on predictions or monitoring of state information. These approaches are detailed next.

MDP. Markov Decision Processes constitute one of the most used approaches to model the migration tradeoff. In this model, a decision maker in a given state is presented with a set of possible actions it can take (e.g. to migrate or not to migrate), each leading to an immediate reward (alternatively penalty/cost). Consequently, it has to decide which action to take in order to maximize its reward (alternatively minimize the penalty/cost) on the long term. Value iteration and policy iteration are the most used approaches to solve the MDP. However, more recently, efforts are instead shifting towards solving the MDP problem using (deep) Q-learning approaches, which do not require prior knowledge about the MDP environment dynamics.

Initial contributions where MDPs have been used for service migration can be found in the context of the Follow-me Cloud in its non-edge-centric version[50, 87]. However, the authors' subsequent work [89] extends this to the case where micro datacenters are deployed at the edge and propose an MDP-based algorithm that is able to capture the tradeoff between the migration cost and the user experience. As shown in the simulation results, the algorithm performs less service migrations when the migration cost is non-negligible compared to the expected quality.

Similarly, authors in [97] model the problem of service migration in mobile micro-clouds (MMCs) as an MDP and derive a modified version of the policy iteration algorithm to obtain the optimal actions to take depending on the distance between the user and its serving MMC. Simulation results show that the proposed algorithm has a low complexity and that it is able to perform migrations when the transmission cost from the previous micro-cloud is high and avoid them otherwise. In [98], authors go beyond the contributions in [97] by additionally considering general cost models, 2D user mobility, as well as using real world mobility traces for evaluation purposes.

Instead of considering only the usual transmission and migration costs, an additional security (i.e. inter-user interference) cost is introduced in [24]. This cost models the risks related to keeping services belonging to different users running in the same/nearby MMCs. Therefore, with an overall objective of reducing the time average costs, the proposed migration policy proactively keeps the services from different users in distant MMCs to reduce future security cost increases. Numerical results show that this policy achieves the lowest sum costs compared to "always migrate", "never migrate" and a "myopic" policy.

In contrast, in [106], authors propose to co-place multiple users of a Virtual Reality- Massively Multiplayer Online Game (VR-MMOG) at the same edge cloud to facilitate sharing their "game worlds" and therefore reduce the migration overhead. Given the complexity inherent to making joint migration decisions rather than per-player decisions, a "Highest Migrate Probability First" heuristic is proposed, where players are ordered according to their migration probabilities. Migration decisions are therefore performed following the obtained players' order. It is shown that the proposed heuristic performs close to the optimal cost.

Instead of solving the MDP problem using complex dynamic programming solutions that require statistical knowledge, authors in [94] leverage Lyapunov optimization to design online algorithms, where prior knowledge about the MDP states is not required. Trace-based simulations prove that the proposed Lyapunov-based algorithm achieves a good tradeoff between the average delay and the cost.

Observing that in previous MDP-based studies, migration decisions do not take into account network and server states, authors in [107] propose an MDP-based QoS-aware service migration where such states are considered. The MDP model's reward function assesses the difference between the incurred downtime and the potential QoS improvement for the user. The implementation of an AR application on the "Open-Access Research Testbed for Next-Generation Wireless Networks

(ORBIT) testbed" along with the use of real-world mobility traces show a reduction in the response time of this scheme compared to the "lowest load" and the "least hop" migration schemes.

Finally, in [91] the problem of container migration in fog computing is modeled as a large scale MDP. Deep reinforcement learning algorithms are designed to reduce the large MDP spaces and achieve faster decision-making. Authors conduct trace-based experiments that confirm the algorithms' ability to obtain reduced delays, power consumption and migration costs.

Markov chains. Authors in [35] consider proactive service replication as an alternative to avoid migration-incurred costs (both in terms of downtime and bandwidth consumption). Since service replication also comes with a storage cost, authors analytically compare the cost incurred by both approaches, taking users' mobility and service duration into account. Results show that for a short-duration service, replication is preferred. On the other hand, migration would be preferred for longer services, since, given their size, they are likely to cause a higher service replication cost.

Mixed integer linear programming (MILP). Authors in [85] propose "PRofIt Maximization Avatar pLacement (PRIMAL)" as a solution to the problem of live Avatar migration in a cloudlet network. More specifically, they consider both the potential migration gain in terms of E2E delay reduction and the migration cost to calculate the migration profit. They prove through simulations that PRIMAL was superior to the "follow me avatar" strategy as well as the no-migration strategy in terms of the migration profit.

Mixed-integer quadratic programming (MIQP). In [101], the problem of VM migration in a "roadside cloudlet-based vehicular network" is formulated as a mixed-integer quadratic programming problem. The aim is to minimize the network cost, while considering the migration-transmission cost tradeoff. In order to solve the formulated problem, authors propose a heuristic that finds, in polynomial time, the optimal network path between a client vehicle and its corresponding VM. Simulations show that the proposed heuristic obtains the lowest network cost compared to state-of-the-art alternatives, while approaching the cost obtained by the optimal solution to the MIQP. Additionally, the obtained results emphasize the need for efficiently balancing both migration and transmission costs.

Stochastic optimization. Authors in [72] study the performance-cost tradeoff that may result from frequent migrations in MEC. A stochastic optimization problem is formulated to minimize the long-term average latency given a long-term cost budget. Due to the lack of knowledge regarding the user mobility path, the original problem is transformed into a set of real-time optimization problems, using Lyapunov optimization. To solve the resulting problems, the Markov approximation technique is used to derive an efficient heuristic. Simulations show that the proposed heuristic achieves the lowest user perceived latency while meeting the defined cost budget.

Multi-Attribute Decision making. In [109], the authors focus on the problem of balancing the costs and benefits related to service migration in MEC by leveraging the "multiple attribute decision making" approach. Simulations show that using this approach significantly reduces the user's response time.

Predictions and monitoring. A different approach for evaluating the tradeoff can be found in some contributions where the benefit-cost information is derived from predictions, generally based on historical data or retrieved from a monitoring module that is constantly updating the relevant state information.

This can be seen in [14], where authors propose to leverage SDN principles to decouple mobility control and data forwarding to achieve service continuity for fog computing users. In fact, the SDN controller installs the new mobility logic either proactively or reactively on the SDN-enabled fog servers. The optimal path to the new fog server is calculated based on the performance gain and system cost, taking into account the predicted user's residence time within the FS's area. Simulations performed using mininet-wifi [34] demonstrate the ability of the proposed scheme to improve the handover performance and the data communication efficiency.

Also leveraging the SDN paradigm, authors in [84] propose a VM migration scheme, which compares the estimated profit resulting from migrating/not migrating the VM, in terms of the amount of traffic reduction in the network core. In fact, the traffic caused by the migration can be predicted based on the average bandwidth available for migration, the average data rate between the user and its VM (both to be obtained from historical traces), the generation rate of dirty memory pages, the VM memory size and the threshold for stopping the iterative copying. On the other hand, the traffic generated from keeping the VM in its originally serving node depends on the average data rate between the user and its VM and the BS retention time, which is also predictable. Trace-based emulation shows that the proposed dynamic migration reduces the network traffic compared to the static VM deployment.

A different approach is provided in [30] where the impact of mobility on QoS in a distributed Regional DataCenters (RDC) context is addressed. More specifically, authors propose an algorithm that evaluates the costs for migrating/not migrating the VM based on information collected from a monitoring module, so that the costs are minimized and the QoS is maintained. Simulations conducted using Mininet show that migrations are not performed by the proposed algorithm when the QoS is met.

Table 5 summarizes the contributions presented in this section, highlighting their main objectives, the approach used to find the optimal solution, the evaluation type, and in the simulation case, whether mobility was based on real world (***RW***) mobility traces or synthetic (***S***) ones, and finally the different factors considered in the migration process.

5.1.2 Mobility pattern-based migration avoidance. In order to avoid excessive migration costs, a number of contributions, summarized in Table 6, rely on predicting future migration targets based on users' mobility patterns.

For instance, to deal with the issue of high migration costs due to frequent state propagation among elements of a fog computing infrastructure, authors in [70] propose to use a time graph structure to model the expected costs. Then, the most suitable migration targets are determined ahead of time, taking users' mobility and dependencies among migrations into account. Evaluation results show substantial savings in network utilization, even in presence of uncertainties characterizing users' mobility patterns.

In [37], authors provide an Integer Linear Programming (ILP) model for VM placement in fog computing. The use of future user position is adopted to improve the VM placement and decrease the number of migrations. Both Myifogsim[53] and SUMO[12] are used to simulate the VM migrations and users' mobility, respectively. Authors find that using information about the following five minutes of the user's position significantly reduces the number of migrations. However, in this work, this information is obtained with a 100% accuracy, which may not apply in real world scenarios.

Finally, in order to meet strict quality requirements of 5G automotive systems, authors in [3] introduce the "Follow Me Edge-Cloud (FMEC)" concept. Within FMEC, a mobility-aware migration is performed to reduce both migration and data transmission costs. More specifically, an algorithm is provided to determine the vehicle's mobility pattern. As a result, if the vehicle is moving fast, the service is migrated to further locations thus avoiding unnecessary migrations along the way. If, on the other hand, the vehicle is decelerating, the migration is performed to the nearest MEC to ensure that delay constraints are met. Both theoretical and simulation-based results show that mobility-aware migration obtains less number of migrations as well as less global cost.

Table 5. Cost-related tradeoffs - Summary

Reference	Model	Objective	Solution approach	Evaluation	Mobility		Considered factors
					RW	S	
[89]	MDP	Find the optimal migration policy capturing the tradeoff between user experience and the migration cost	Value iteration	Simulation		✓	✓Hop distance from serving DC ✓Migrated content size ✓Signaling messages size
[97]	MDP	Minimize the total cost over time	Modified Policy Iteration	Simulation		✓	✓Transmission cost ✓Migration cost
[98]	MDP	Find a policy that minimizes the long-term expected total cost	Modified Policy Iteration	Simulation	✓		✓Migration cost ✓Transmission cost
[24]	MDP	Minimize the total time-averaged costs over the network	Modified myopic policy	Simulation		✓	✓Migration cost ✓Transmission cost ✓Security cost (multi-user interference)
[106]	MDP	Minimize long term cost	Heuristic	Simulation		✓	✓Migration cost (in terms of the transferred size) ✓Transmission cost: with and without mutual impact among users
[94]	MDP	Minimize cost subject to average delay constraint	Online algorithm	Simulation	✓	✓	✓Average delay constraint ✓Queuing cost at edge cloud ✓Backend transmission cost ✓Migration cost
[107]	MDP	Maximize long term reward (measuring the tradeoff between the QoS improvement and the migration cost)	Value Iteration	Simulation	✓		✓Migration cost (service downtime) ✓Predicted QoS ✓Request frequency (reflecting the load)
[91]	MDP	Minimize costs	Deep Q-learning	Testbed	✓		✓Network delay ✓Computation delay ✓Power consumption ✓Migration cost
[35]	Markov Chains	Evaluate the tradeoff between replication costs and migration costs	-	Simulation			✓Replication cost (storage) ✓Migration cost ✓Service duration ✓Residence duration
[85]	MILP	Maximize the total Avatar migration profit (the profit captures the tradeoff between the migration gain and the migration cost)	Heuristic derived from CPLEX solver	Simulation		✓	✓Migration time ✓Application type (I/O intensive…) ✓BS-cloudlet delay
[101]	MIQP	Minimize total network cost	Heuristic	Simulation		✓	✓Cost in terms of number of hops between old RSC and new RSC and between vehicle and new RSC
[72]	Stochastic Optimization Problem	Minimize user-perceived latency subject to a long-term migration cost budget	Heuristic	Simulation		✓	✓Computing delay ✓Communication delay ✓Migration cost
[109]	-	Find an efficient strategy that decides whether, when and where to migrate	Multi-attribute decision making	Simulation		✓	✓Bandwidth ✓Computing ability ✓Latency ✓Costs (migration and communication) ✓Energy consumption
[14]	-	Find an optimal path to the new fog server	Predictions	Simulation		✓	✓Performance gain based on predicted residence time ✓Signaling cost
[84]	-	Reduce traffic in the SDN-based cellular core using dynamic VM migrations	Predictions	Simulation	✓		✓Migration traffic cost (Average bandwidth, data rate between user and VM, VM memory size, pre-copy related parameters) ✓Transmission traffic (user-VM average data rate) ✓BS retention time
[30]	-	Maintain the QoS while supporting service mobility among distributed regional datacenters	Predictions	Simulation & Testbed		✓	✓Migration cost (computing cost at the source and destination due to migration, bandwidth) ✓Transmission cost(WAN bandwidth cost, computing cost at the source node) ✓VM migration time

Table 6. Mobility pattern-based migration avoidance - Summary

Reference	Objective	Evaluation	Mobility		Considered factors
			RW	S	
[70]	Reduce network utilization by planning migrations ahead of time while ensuring E2E latency is met	Simulation		✓	✓User mobility pattern ✓Dependencies among migrations
[37]	Improve VM placement and decrease the number of migrations	Simulation	✓		✓Cost (in terms of latency) ✓Future user position
[3]	Provide a mobility-aware service migration	Theoretical & simulation		✓	✓Minimum QoS requirement (delay) ✓User velocity ✓Migration costs ✓Transmission costs

5.2 Time-related optimizations

Migration optimizations focusing on the time axis can be further divided into optimizations proposed at the virtualization technology level, optimizations leveraging Multipath TCP (MPTCP), optimizations resulting from taking proactive actions prior to effective migration and optimizations to the service re-assignment process.

5.2.1 Virtualization technology-related optimizations. Contributions in this category assume the decision on "when and where to migrate" has been reached using one of the approaches discussed in Section 5.2.1. As such, they rather focus on reducing the downtime as well as the total migration time, as we detail next.

VM-related optimizations. Authors in [108] propose to improve the default strategy for dirty page transfer in Xen[10], where pages that have been previously sent are sent again in the following iterations. They instead propose a strategy that copies a fixed number of pages in each round. These pages are selected as the ones with the smallest dirtiest rates. The same problem is addressed in [69], where a "smart pre-copy VM live migration framework" is proposed. The amount of dirty pages observed in previous iterations is provided as an input to a regression model that predicts the expected downtime in each iteration. Then, if the predicted downtime is greater than a predefined downtime threshold, the "stop and copy" stage can be performed. Although authors claim that the proposed framework can ensure minimum downtime, no evaluation results have been provided. Aligned with the previous two proposals, authors in [21] propose a new pre-copy algorithm, where "cold" pages, i.e. the memory pages with lower probabilities of being written to are sent to the destination while the "hot" pages are skipped, hence the proposed name "hot-skip". This results in a reduced length of iterations and a shorter migration time, as shown in the conducted simulations.

A different approach to reduce VM migration delays is adopted in [40, 41, 102]. In fact, authors in [102] propose two VM migration approaches. In the first one, termed "full migration", a snapshot of the VM is taken by the old fog. After compressing the resulting data, the old fog sends it to the target fog, where it will be decompressed and resumed. An enhancement to this lies in the second approach, i.e. the "incremental migration", which considers that both fogs have the base VM image, therefore, only the incremental part will be sent to the destination fog, thus resulting in shorter response times. Authors in [40] propose a pipeline comprised of different stages, aiming to optimize the state transfer between the source and the destination cloudlets, by efficiently capturing the state differences among them, while in [41], pre-caching base VM images into every cloudlet is performed so that blocks which are already present at the target do not need to be transferred again.

Container-related optimizations. Despite constituting a promising step towards alleviating the downtime issue, the afore-mentioned VM-centric approaches fail to meet the delay constraints of certain time-sensitive applications. This has led the research community to transition from VM-specific optimizations to container-specific ones. For instance, authors in [56, 57] propose a three-layer framework for migrating active service applications in a MEC environment. The proposed framework applies for containers (but also to VMs) and is based on the use of three layers, a generic "base layer" present in all MECs, an "application layer" that consists in "an idle version of the application" and an "instance layer" that contains the running state. Using this framework, only layers which have not been found at the destination will be transferred and therefore significant time savings can be achieved. Similarly, building on the layered structure of Docker containers, authors in [54, 55] propose avoiding the transfer of redundant image layers to reduce latency. A different approach is proposed in [38] where "redundancy migration" is proposed to avoid the time-consuming "stop and copy" live migration stage. This approach works as follows. As soon as the destination server (DS) is selected, the user starts buffering its packets to it. It is then up to the DS to forward these packets to the old sever and retrieve the results to be sent back to the user. Then, while the application is still running in the old server, its checkpoint is created and sent to the DS. The DS uses this checkpoint to restart the container and starts replaying buffered packets from the moment of the checkpoint creation. When the buffer is empty, the destination container takes over and the source container can be shut down. Evaluation results conducted against LXD live migration show that "redundancy migration" is able to effectively reduce the latency. Authors in [1] address the need for a lightweight migration of service states among MECs by proposing three different mechanisms of container-based live migration. They propose an iterative approach and a diskless approach in the case where the user path can be known, whereas a shared storage-based approach is proposed for situations where the mobility pattern is random. Testbed experiments are conducted to compare the three approaches and show their performance in terms of migration time and downtime reductions.

Process-related optimizations. Apart from optimizations done at the VM and container levels, process-level optimization has been proposed in [45] and [19]. Motivated by the need to minimize network resource usage in edge environments, the main idea in [45] consists in only transferring stateful processes that contain application states; the OS and the stateless processes do not need to be migrated. In order to enable communication continuity between the processes, authors introduce a method to properly convert inter-process communication channels. The prototype implementation results show less downtime compared to VM and container migrations, in addition to a low overhead of the proposed channel conversion. Process-level optimization is also considered in [19], where authors present multi-context processes (MCPs) as an additional degree of virtualization other than usual techniques (e.g. bare-metal, hypervisor-based, container-based). An MCP can be seen as a sort of workspace that runs directly on the hardware, bypassing the hypervisor. A possible implementation of MCPs can be done using DPDK libraries[28]. VMs and MCPs differ in the size of their virtual (base) images. Authors also consider that these virtual images are present at the destination. Therefore, only the dynamic state needs to be transferred in order to reduce the overhead. Evaluation results show that the migration downtime for VMs is over 2 orders of magnitude greater than MCPs.

Bare-metal-based. A different idea is presented in [5], where authors advocate the vision for "bare-metal edge computing". Following this approach, system administrators can perform a lightweight, ARM-based, "bare-metal live migration". The goal would be to increase the service execution performance while consuming less energy. The authors primarily focus on the definition of challenges for the realization of such a vision and as such, no evaluation results have been provided.

Table 7 summarizes the presented contributions in terms of the used technology, as well as the evaluation approach, if any.

Table 7. Virtualization technology-related optimizations - Summary

Reference	Technology				Evaluation
	VM	Container	Process	Bare-metal	
[108]	✓				Simulation
[69]	✓				-
[21]	✓				Simulation
[102]	✓				Testbed
[40]	✓				Implementation
[41]	✓				Implementation
[56], [57]	✓	✓			Emulation
[55], [54]		✓			Implementation
[38]		✓			Implementation
[1]		✓			Testbed
[45]			✓		Prototype
[19]			✓		Testbed
[5]				✓	-

5.2.2 Multipath TCP-based optimizations. Another approach to reduce the overall migration delay consists in using Multipath TCP(MPTCP). MPTCP relies on the simultaneous use of multiple interfaces, thus allowing to send data across different subflows[63]. This is particularly beneficial to improve the resilience of the migration in edge-centric environments, since migration is performed over WAN links that may suffer from congestion or failures issues[78].

One of the relevant works in this area can be found in [92], where the live VM migration across WAN-connected cloudlets is considered. The proposed migration approach combines MPTCP with the pre-configuration of the VM IP address at the destination. Experiments on Linux KVM[49] show that the proposed approach eliminates the service downtime. This approach has been also used in [21] to parallelize the VM migration, leveraging the aggregate throughput provided by MPTCP. Apart from VM migration, MPTCP has been also proposed for LXC container migration in [78], where the experimental results show the efficiency of this method in reducing the migration times.

5.2.3 Proactive actions. Reducing migration latency has been also addressed in the literature by proactively taking actions to prepare the migration environment or by planning the migration execution. The relevant contributions in this category are described next and summarized in Table 8.

Authors in [13] enhance the Elijah edge computing platform[42] with a proactive service migration implementation. Within this implementation, users' movements are predicted using a regression model. Then, when the user is predicted to be changing its serving MEC, the target MEC is requested to initiate the migration procedure. Authors obtained a VM downtime of 1.60s thanks to the proactive approach. Aligned with this idea , the "follow me edge" (FME) concept has been introduced in [86] with an aim to "ensure that the user is always serviced from the closest edge". Within FME, the user's location is used to estimate the latency to both the serving and the target edge servers, and if appropriate, migration is triggered proactively.

In [96], a different approach is taken to reduce service migration latency across cloudlets. The proposed approach considers the case of an AR application and therefore leverages the features extracted from the user's camera to predict the target cloudlet prior to the radio handoff. Experiments conducted on a testbed show around 65% reduction in migration latency compared to the reactive scheme.

Authors in [44] consider mobile situation awareness applications that require low-latency processing of sensor data events so that live situational information can be delivered to users in time. To meet such low latency requirements, the proposed system ensures that sensor events are proactively processed in those regions along the user's mobility path where the request for situational information is likely to occur. Simulations confirm that doing so significantly reduces the perceived latency.

The Follow Me Fog (FMF) framework has been proposed in [8]. In FMF, computation jobs are pre-migrated from one fog node to the other, prior to the the wireless handover. Results obtained from a prototype implementation show substantial latency reductions are achieved with the FMF pre-migration strategy. Building on FMF, sFog has been proposed in [9] where a more detailed handover protocol, the corresponding theoretical framework as well as a congestion control scheme were added compared to FMF.

In [32], a proactive service migration framework has been designed, with a specific focus on stateless applications in MEC environments. The main idea consists in placing service replicas in neighboring edge nodes so that service instances are ready to be used by the user when the handover occurs. Emulation shows that this proactive approach has a shorter migration time and a reduced amount of transferred data, compared to its reactive counterpart.

As opposed to approaches listed in Section 5.1.2, where planning migrations ahead of time had the objective of reducing the number of costly migrations, we identify in the following some works where migrations are planned beforehand for time saving purposes. These include the "mobility-based services migration prediction (MSMP)" scheme proposed in [65]. MSMP takes into account the user's mobility pattern as well as the expected load of the MDCs and then plans the processing and migration of the different portions of the service within MDCs in the user's path. Latency reductions have been observed in simulations thanks to the efficient planning achieved through the MSMP scheme. On the other hand, authors in [110] consider a MEC scenario where the Virtualized Services (VSs) migration problem has been formulated as an integer programming problem with an objective to maximize the service availability, while keeping the migration time below a pre-defined value. A low-complexity heuristic has been proposed to calculate the migration schedules. Simulations show that the proposed heuristic was effectively able to obtain a near optimal performance.

5.2.4 *Time-related optimal service re-assignment.*

This section reports on the contributions that address the problem of finding the optimal node to migrate the service to, with an objective or a constraint related to the time axis.

As an example, authors in [46] developed a mobility- and load-aware virtual machine migration (VMM) for a mobile cloud computing context. This is achieved by using a genetic algorithm (GA) to select the optimal cloudlet to migrate the VM to such that the total number of VM migrations is minimized, resulting in a reduced task execution time. Simulation results confirm that the GA-VMM approach achieves the lowest time for task execution and the lowest number of VM migrations, compared to the no-migration case, load-centric migration as well as greedy migration.

Focusing on a fog computing scenario instead, authors in [33] propose a community-based approach to address the problem of service placement and migration. In this work, where the concept of "community" originates from the field of cloud resource management, the network of

Table 8. Proactive actions - Summary

Reference	Objective	Evaluation	Mobility		Factors
			R	S	
[13]	Reduce migration-related delays	Implementation	-	-	✓Predicted user movement
[86]	Ensure that Content follows the physical mobility of users to prevent quality degradation	Testbed	-	-	✓User location
[96]	Reduce migration latency between cloudlets	Testbed	-	-	✓Features extracted from the user's camera
[44]	Ensure live situational information is ready to be used by the time the user reaches its next location	Simulation		✓	✓User location ✓Temporal interest ✓Spatial interest
[8][9]	Seamless service execution in case of mobility through job pre-migration	Implementation	-	-	✓RSS conditions ✓Job status
[32]	Provide a proactive service migration for stateless applications in MEC	Testbed	-	-	✓Geographical proximity ✓Application requirements
[65]	Plan, for a given service, the assignment of different portions of the service to the micro data centers	Simulation	✓		✓Known user path ✓Residence time ✓Traffic load
[110]	Maintain service continuity when migrating virtual services	Simulation	-	-	✓Target migration time ✓Service importance ✓MEC capacity

fog nodes is partitioned into a hierarchical structure, where communities are combined to cover wider geographical regions, as we move higher into the hierarchy. For evaluation purposes, the notion of communities has been integrated into the CloudSim simulator[20] . The obtained results show the ability of the community-based proposal to follow the mobile user's movement, thus resulting in a close-to-optimal average service delay.

Authors in [18] propose a clustering and migration policy for virtual objects (VOs) that constitute cloud and fog services. This policy not only considers the need for meeting user proximity requirements but also takes inter-VO affinity (dependencies) into account. Simulation results show the QoS improvement when migrations are performed and the potential of considering VO clustering in reducing the number of migrations.

A vehicular fog computing environment is considered in [112] where both clients and fog nodes are mobile. A MILP-based optimization is performed to select a new fog node for the task execution such that the E2E latency and the quality loss are balanced. A reduction in the service latency is observed in the trace-based simulations results.

Also considering a vehicular scenario, authors in [100] provide an algorithm to determine the time at which an application should be switched to a new cloudlet by considering link durations among bus-hosted cloudlets. Then, the best cloudlet is selected as the next application host such that the energy consumption of the requesting mobile device is minimized, while ensuring the application delay constraint is met. Simulation results confirm that the proposed scheme efficiently meets this objective.

Focusing on the context of Green Cloudlet Networks (GCN), i.e. cloudlet networks "powered by both green and brown energy", authors in [31] propose the "Energy driven AvataR migration (EARN)" scheme. The problem is formulated as a MILP problem that considers the energy consumption cost caused by the Avatar migration and aims to minimize the total "brown" energy consumption such that the UE's SLAs in terms of delay are met. The suboptimal solution is obtained using the branch and cut algorithm, which ensures that less Avatar migrations are performed when the green energy generation at the cloudlet is low.

Finally, authors in [23] consider an environment where the cloud, the cloudlets, human users and collaborative robots (cobots) on the move, collaborate in real-time to execute a set of tasks. Within this environment, a context-aware task migration scheme is proposed. Several factors are taken into account in the migration decision, including the "processing capabilities of cloud/cloudlet agents and cobots, the task execution deadline, the energy consumption of the involved cobots and the mobile devices, and the task migration latency"[23]. Multiple performance metrics are provided to confirm the benefits of the proposed task migration compared to a "no migration" approach.

Table 9. Time-related optimal service re-assignment - Summary

Reference	Objective	Solution approach	Evaluation	Mobility		Considered factors
				RW	S	
[46]	Select optimal cloudlet for the VM such that number of migrations and the task execution time are reduced	Genetic algorithm	Simulation		✓	✓Load ✓VM transfer time ✓Task completion time
[33]	Address the placement and migration problems to provide a "follow me" experience to the end users	Community concept	Simulation		✓	✓User location ✓Resource availability
[18]	Scalably support user mobility in a Cloud-Fog environment by migrating clusters of virtual objects	Graph-based	Simulation		✓	✓User class ✓Proximity requirements (latency, path length) ✓Inter-dependencies
[112]	Select new FN for task execution such that the E2E latency and the quality loss are balanced	MILP-based	Simulation	✓		✓Maximum service latency ✓Tolerance of quality loss ✓FN capacity ✓Processing delay ✓Transmission latency
[100]	Select next cloudlet to migrate to such that energy consumption of the mobile device is minimized while satisfying application delay constraint	Algorithm	Simulation		✓	✓Inter-cloudlet link duration ✓Cloudlet-requestor link availability ✓Required completion deadline ✓Device energy consumption
[31]	Migrate Avatars to green energy-powered cloudlets while meeting users' latency requirements	Algorithm (Branch and cut)	Simulation		✓	✓SLA (in terms of E2E delay) ✓Energy consumption caused by the migration at the source and destination ✓Energy consumption for Avatar execution
[23]	Provide a context-aware task migration scheme	Algorithm	Simulation		✓	✓Processing capabilities ✓Task execution deadline ✓Energy consumption ✓Migration latency

5.3 Migration success rate optimization

Contributions falling under this category originate exclusively from the vehicular cloud research area. This is mainly due to the dynamicity characterizing such a scenario, thus potentially negatively affecting the migration success rates.

More specifically, authors in [104] address the problem of unsuccessful migrations that may arise when the target cloud is unable to accommodate the VM, because it is excessively loaded. A resource reservation scheme is proposed to address this issue and an optimization problem is formulated to derive the optimal number of reserved resources. The proposed reservation scheme was shown to be promising, as a reduction of the service dropping rate was observed in the simulation results.

To cope with the problem of vehicles hosting services leaving their area of operation before the end of the service execution, several "Vehicular Virtual Machine Migration (VVMM)" schemes have been proposed in [79]. Such schemes ensure that the VM is migrated from the exiting vehicle to other vehicles remaining in the RSU area. The most promising scheme is the "Mobility and

Destination Workload Aware Migration (MDWLAM)" scheme, which obtained very low migration drop rates, as shown in the conducted simulations.

In line with this approach, authors in [64] leverage mobility prediction to avoid workload losses when a vehicle moves to another RSU area or a non-covered area. The vehicle's lifetime within the current RSU area is predicted based on an Artificial Neural Network (ANN) model, and when a predefined portion of its lifetime elapses, its current workload (i.e. its corresponding VM) would be pre-migrated to another vehicle that has a long lifetime in that same RSU area.

Authors in [11] study the feasibility of performing VM migrations in a VANET scenario using V2V communications, instead of using cellular links. However, since V2V communications are intermittent, authors propose to opportunistically perform migrations in hotspot areas, defined as areas "where vehicles come in contact more often and for [a] longer period". Doing so improves the ratio of successful migrations, as shown in trace-based simulations.

Table 10. Migration success rate optimization - Summary

Reference	Objective	Evaluation	Mobility		Considered factors
			R	S	
[104]	Minimize dropping rate during VM migration	Simulation		✓	✓Storage resources ✓Computing resources
[79]	Minimize migration drop rates	Simulation		✓	✓Predicted residence time ✓Destination workload ✓Expected migration duration
[64]	Avoid VM workload losses due to mobility of hosting vehicles	Simulation		✓	✓Predicted lifetime
[11]	Improve ratio of successful migrations	Simulation	✓		✓Contact duration ✓Migration duration

6 ARCHITECTURES, PLATFORMS, SIMULATORS AND DEMONSTRATORS

As opposed to works reported in Section 5, where the focus was targeted towards meeting certain system-level objectives, this section highlights the works where the main objective was to design and develop architectures, platforms, simulation tools and demonstrators to support service migration in edge-centric computing environments.

6.1 Architectures

One of the early architectures for VM migration in fog computing has been proposed in [15]. One key functionality within this architecture is the "Mobility behavior and handoff analysis" which determines the right time to perform a migration as well as the migration target based on users' movement. This is further facilitated by the actual VM/container migration module as well as the cloudlet discovery module, where the fog topology is constructed. Authors in [95] present an architecture for service placement in the IoT. In fact, after the initial placement, services can be continuously migrated to cope with changing users' and network conditions. An ILP formulation is proposed with the aim of minimizing (i) the hop count between users' and serving nodes, (ii) the hop count between nodes in case services need to communicate due to dependencies linking them, and finally (iii) the number of migrations to reduce oscillations in the system. More recently, Companion Fog Computing (CFC) has been proposed in [75] to enforce that a container running a fog service always remains topologically close to the mobile IoT device. A reference architecture along with its migration-related functionalities are described. An implementation based on the S4T platform[17] is provided and performance metrics including the downtime, the migration time and the memory footprint of the container states were derived.

Considering a cellular network context instead, authors in [4] define an architecture for service mobility between two eNodeB-clouds, leveraging standard protocols. A prototype implementation of this architecture is developed and reasonable context migration times were observed. Authors in [66] consider the problem of the costs incurred by frequent handovers and service migrations, when considering densely-deployed femtocells having MEC capabilities. The authors first propose an architecture that introduces a SharedMEC entity that facilitates sharing of migration information among a group of Femto-BSs. Then, this shared information is leveraged by a service handover decision algorithm to make the appropriate migration decisions. A set of simulations have been conducted and the use of the SharedMEC entity along with the service handover algorithm was shown to achieve the desired cost reductions. Authors in [80] consider a MEC-enabled 5G architecture. An NFV Orchestrator (NFVO) entity ensures that application and network virtual network functions are properly migrated. A testbed implementation is introduced to prove the feasibility of the considered architecture.

Finally, the work in [22] introduces an architecture named Edge Cognitive Computing (ECC) that "combines edge computing and cognitive computing", where cognitive computing provides machines with ""brain-like" cognitive intelligence". Within this context, the authors propose a reinforcement learning-based service migration mechanism, that addresses the tradeoff of cost minimization and QoE improvement. Experimental results show that the proposed ECC architecture can provide higher QoE compared to scenarios where cognition is not considered.

6.2 Platforms

Apart from architectures, multiple platforms have been developed to validate different migration approaches.

One such platform is Foglets, presented in [83]. Foglets provides APIs for application development as well as mechanisms for application component deployment. Additionally, QoS-sensitive and workload-sensitive migration mechanisms are proposed to deal with mobility and application dynamics. The obtained results show that QoS-sensitive proactive migration can be accomplished in 6ms in an emulated environment.

Authors in [29] introduce Cloud4IoT as a "platform able to containerize IoT functions and optimize their placement". Both gateway-to-gateway (i.e. horizontal) and cloud/edge-to-gateway (i.e. vertical) migrations are included. To fully support its envisioned functionalities, Cloud4IoT uses relevant state of the art tools such as Openstack[68] and Kubernetes[51] for cloud and container management.

Authors in [47] propose a "service management platform that supports migrating IoT services" in a fog computing architecture. Its evaluation has been carried out in an emulated setup that showed a latency reduction compared to static edge- or cloud-based service placements.

Finally, authors in [27] present an edge computing platform based on Openstack++[42]. The proposed platform orchestrates container migration using MQTT and then uses CRIU[26] for the effective migration. The use of MQTT is justified by its ability to perform asynchronous communications, thus achieving minimal usage of the network bandwidth, as needed in an edge environment.

6.3 Simulation tools

One relevant simulation tool that is worth-mentioning in the migration context is MyiFogSim[53]. This simulator builds upon iFogSim[39], a recently-proposed fog computing simulator, and enhances it with the VM migration feature to cope with the inherent mobility characterizing fog environments.

Along with MyiFogSim, [53] proposes a migration policy to determine the timing for starting the migration. This is achieved by monitoring the user's position, its speed and the direction of movement and verifying this information against previously known AP positions. MyiFogSim is then used to derive results showing that lower latencies can be obtained using the proposed policy, compared to a "no migration" scenario.

In addition to MyiFogSim, FogNetSim++ has been proposed in [77] as an extension to OMNet++ to simulate fog computing environments. FogNetSim++ provides a feature for performing handovers among fog nodes, either due to the device's mobility or for load balancing purposes. However, support for VM migration in FogNetSim++ is presented as a future work.

6.4 Demonstrators

In this section, we list some of the demonstrators developed to obtain real-world evaluations of certain migration scenarios.

The demonstration in [71] shows, through a small-scale fog computing deployment, the importance of proactively moving operators (See definition in Table 2) from one broker to the other following the user movement. An example video monitoring application is used to show that improper placements and migrations lead to a poor quality of experience.

Aligned with this, authors in [82] highlight an example usage scenario of a the Foglets framework[83] presented earlier. More specifically, they show how context-aware migrations can be performed based on the user's position. This leads to a seamless service execution, as was observed for the considered video streaming application.

Authors in [48] provide an SDN-based implementation of a cyber-function migration from one location on the cyber infrastructure to the other due to drone movement. Using Linux containers to implement the cyber functions, a migration downtime of 1.6s has been observed.

Table 11 provides a summary of the aforementioned contributions.

7 GAP ANALYSIS AND RESEARCH OPPORTUNITIES

As shown in the previous sections, the surveyed literature revealed to be rich on many levels. However, some gaps could be noticed, turning into potential research opportunities that we summarize next:

- **Decision-making placement:** Apart from a few exceptions, most of the reviewed contributions dealing with the migration decision making do not explicitly state where the decision-making logic is placed, and most importantly do not discuss its impact on the migration performance. Three potential decision-making placement levels could be envisioned as suggested in [36] for a mobile cloud computing context. These include centralized decisions, server-level decisions and task-level decisions. As we go from the former to the latter, the autonomy level increases whereas the complexity decreases. However, apart from autonomy and complexity, other considerations such as the impact of the placement on the timeliness of the decision and the availability of the required decision making information at the considered level also matter. We find that only authors in [91] reiterate the need to address such a problem.
- **Migration-related synchronization traffic:Timing and overhead:** To enable a more informed migration decision-making, migration-related information such as the mobility path and load status information needs to be exchanged periodically among fog nodes. However, the impact of the periodicity timing and the resulting overhead in terms of generated network traffic has not been studied much. For instance, authors in [98] state that this information exchange does not occur frequently, thus leading to a low overhead. However, this has not

Table 11. Architectures, platforms, simulators and demonstrators - Summary

Reference	Type	Description	Considered factors	Evaluation
[15]	Architecture	General architecture supporting VM migration in fog computing	✓Users' movement	-
[95]		Architecture for service placement and migration	✓Hop count between users' and serving nodes ✓Inter-node hop count	-
[75]		Reference architecture comprising migration-related functionalities	✓Topological distance ✓Maximum tolerated topological distance ✓Hardware requirements ✓Resource availability ✓Data protection levels	Implementation
[4]		Architecture for user context migration between EnodeB-Clouds	✓Handover requests	Implementation
[66]		Architecture to support user mobility by employing a SharedMEC entity	✓Available resources ✓Application type ✓Application availability at the destination	Simulation
[80]		MEC-enabled 5G architecture with support for network and application VNF migration and placement	✓Application requirements ✓Available resources	Implementation
[22]		An edge cognitive computing architecture combining edge computing and cognitive services with support for service migration	✓User behavior ✓Network information	Implementation
[83]	Platform	Distributed programming platform with features for discovery, application collocation, communication APIs and migration	✓QoS constraints ✓Load	Implementation
[29]		Platform for performing horizontal and vertical migration of IoT functions.	-	Implementation
[47]		Service management platform supporting migrations	✓Communication costs ✓Hop count ✓Throughput ✓Latency ✓Resource capabilities	Emulation
[27]		Edge computing platform with container migration orchestration	-	Implementation
[53]	Simulation tool	Extension to the iFogSim simulator to support user mobility through VM migrations	✓User position ✓Speed ✓Direction of movement	Simulation
[77]		Simulator for fog environments with support for handovers among FNs	✓Mobility model ✓Device location ✓Task size	Simulation
[71]	Demonstrators	Show the impact of proactive migration on the QoS in the context of a video monitoring application	✓Future target prediction	Implementation
[82]		Perform application state migration in a context-aware manner	✓Relative user position	Implementation
[48]		Perform cyber-functions migration to an appropriate location on the cyber-infrastructure	-	Implementation

been evaluated in a real-world context. Yet, this aspect is of an extreme importance in the context of fog computing, which was envisioned to alleviate the burden on the network and to guarantee timely service provisioning to end users.

- **User mobility: To predict or not to predict:** While there are several studies that push towards predicting future user positions to pre-migrate the services to the nodes placed in locations that the user is likely to visit, other studies consider that users' mobility is generally difficult to predict and is characterized with inherent uncertainty. To this end, they rely on online learning mechanisms leveraging recent advances in reinforcement learning.

Certainly, each approach can have a set of advantages and disadvantages. For instance, when predictions are obtained with a high accuracy, considerable time savings can be achieved, which is beneficial for latency-sensitive edge applications. However, when prediction errors occur, this approach can result in non-negligible losses that may affect the applications' performance. On the other hand, in online learning, learning can be performed through trial and error following consecutive interactions with the considered environment. However, converging to optimal decisions may require a lot of iterations, which may not be tolerated in time-constrained applications, which are typical in fog/edge computing environments. An appropriate combination of both approaches may therefore be needed.

- **Green energy:** In line with what has been presented in [31] and [84], including green energy efficiency in the migration decision criteria, would be beneficial to achieve a reduced on-grid energy consumption, thus resulting in increased sustainability.
- **Federation:** The problem of performing migrations across federated fog domains has not been explicitly addressed in the literature and most of the reviewed works implicitly consider a scenario of a single provider. However, if migration across federated domains run by different providers is to be considered, several challenges have to be addressed, mostly related with SLA enforcement mechanisms, as suggested in [74]. This not only involves user-provider agreements, but also agreements among different providers with potentially conflicting interests. To this end, deriving solutions inspired from roaming in the telecommunications industry may be envisioned as a starting point for facilitating migrations across multi-provider fog domains.
- **Joint radio handover-service migration optimization:** Joint radio handover-service migration decision optimization could be further investigated in situations where this is relevant, especially with the advent of 5G. This could not only enhance the user's service performance, but it is also likely to result in a more efficient usage of the network and the edge computing infrastructure. So far, only a few contributions, such as [66] and [96], have addressed these aspects in a joint manner.
- **Mobile user-mobile FN scenario:** Even though the mobile user-mobile FN scenario is likely to occur frequently in the context of fog computing, especially with the advent of smart vehicles that can act as fog nodes, this scenario has not been explored much in the literature, as pointed out in Section 3. In fact, even the works that deal with mobile vehicles as VM hosts do not explicitly specify whether the clients, which are being served by those vehicles, are also mobile or not. Further optimizations in this regard can be made considering both clients' and vehicles' mobility patterns, especially when dealing with specific types of vehicles (e.g. buses) having relatively stable routes.
- **Standardization and implementations:** As stated in [32], the ETSI MEC specification provides guidelines for application relocation between Mobile Edge Hosts (MEHs) in MEC. Such a specification can be found in [61], where a set of requirements, use cases and issues pertaining to mobility in MEC have been outlined. However, no such guidelines have been reported in the context of fog computing by the OpenFog consortium (OFC) [67]. In fact, the OFC reference architecture emphasizes that application migration across nodes spanning different levels of a fog deployment should be supported, but no additional details with this regard were provided.

  On the technology level, since the use of the container technology is increasingly gaining in popularity for different service implementations, efforts towards implementing efficient live container migration techniques, such as the efforts done in the P.Haul project[73] should be further reinforced. This will substantially reduce the migration time for time-critical edge applications.

8 CONCLUSION

With the advent of the fog computing and related edge-centric paradigms, users' applications characterized with tight latency requirements and high computational needs are pushed from traditional cloud data centers to edge-based infrastructures to meet the expected requirements. However, given the mobility concerns (affecting users, edge devices and potentially also FNs), running services might need to be migrated from the old serving node to another one, which is closer to the new user location, in order to maintain an optimal quality. This has led to an increasing interest in the topic of service migration at the edge, where a rapidly evolving contributions' landscape is starting to emerge. While certain commonalities can be found within this landscape, such as the need for near zero downtimes, a seamless service execution as well as lightweight migration techniques to cope with the capacity limitations inherent to edge environments, there are also different perspectives from which the migration problem has been tackled. In light of this, this paper has provided a holistic view on these different perspectives, while encompassing literature from different edge-centric research areas. This ensures that advances from different research areas and different perspectives can be complementary and when combined, can contribute to the realization of efficient migration solutions in real edge/fog computing implementations.

ACKNOWLEDGMENTS

This work was partially supported by the Spanish Ministry of Economy and Competitiveness and the European Regional Development Fund under contract TEC2015-66220-R (MINECO/FEDER), and by the H2020 EU mF2C Project ref. 730929.

REFERENCES

[1] Rami Akrem Addad, Diego Leonel Cadette Dutra, Miloud Bagaa, Tarik Taleb, and Hannu Flinck. 2018. Towards A Fast Service Migration in 5G. In *2018 IEEE Conference on Standards for Communications and Networking (CSCN)*. 1–6. https://doi.org/10.1109/CSCN.2018.8581836

[2] Raja Wasim Ahmad, Abdullah Gani, Siti Hafizah Ab Hamid, Muhammad Shiraz, Abdullah Yousafzai, and Feng Xia. 2015. A survey on virtual machine migration and server consolidation frameworks for cloud data centers. *Journal of network and computer applications* 52 (2015), 11 – 25. https://doi.org/10.1016/j.jnca.2015.02.002

[3] Abdelkader Aissioui, Adlen Ksentini, Abdelhak Gueroui, and Tarik Taleb. 2018. On Enabling 5G Automotive Systems Using Follow Me edge-Cloud Concept. *IEEE Transactions on Vehicular Technology* (2018), 5302–5316. https://doi.org/10.1109/TVT.2018.2805369

[4] Hani Assasa, S V Yadhav, and L Westberg. 2015. Service Mobility in Mobile Networks. In *2015 IEEE 8th International Conference on Cloud Computing*. 397–404. https://doi.org/10.1109/CLOUD.2015.60

[5] Ilias Avramidis, Michael Mackay, Fung Po Tso, Takaaki Fukai, and Takahiro Shinagawa. 2018. Live migration on ARM-based micro-datacentres. In *Consumer Communications & Networking Conference (CCNC), 2018 15th IEEE Annual*. 1–6. https://doi.org/10.1109/CCNC.2018.8319241

[6] Victor Bahl. 2015. emergence of micro datacenter (cloudlets/edges) for mobile computing. Retrieved December 16, 2018 from https://www.microsoft.com/en-us/research/wp-content/uploads/2016/11/Micro-Data-Centers-mDCs-for-Mobile-Computing-1.pdf

[7] Ahmet Cihat Baktir, Atay Ozgovde, and Cem Ersoy. 2017. How can edge computing benefit from software-defined networking: a survey, use cases, and future directions. *IEEE Communications Surveys & Tutorials* 19, 4 (2017), 2359–2391. https://doi.org/10.1109/COMST.2017.2717482

[8] Wei Bao, Dong Yuan, Zhengjie Yang, Shen Wang, Wei Li, Bing Bing Zhou, and Albert Y Zomaya. 2017. Follow Me Fog: Toward Seamless Handover Timing Schemes in a Fog Computing Environment. *IEEE Communications Magazine* 55, 11 (2017), 72–78. https://doi.org/10.1109/MCOM.2017.1700363

[9] Wei Bao, Dong Yuan, Zhengjie Yang, Shen Wang, Bing Zhou, Stewart Adams, and Albert Zomaya. 2018. sFog: Seamless Fog Computing Environment for Mobile IoT Applications. In *Proceedings of the 21st ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. 127–136. https://doi.org/10.1145/3242102.3242107

[10] Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt, and Andrew Warfield. 2003. Xen and the art of virtualization. In *ACM SIGOPS operating systems review*, Vol. 37. 164–177. https://doi.org/10.1145/945445.945462

[11] Benjamin Baron, Miguel Campista, PromÃĺthÃĺe Spathis, Luis Henrique M K Costa, Marcelo Dias de Amorim, Otto Carlos M B Duarte, Guy Pujolle, and Yannis Viniotis. 2016. Virtualizing vehicular node resources: Feasibility study of virtual machine migration. *Vehicular Communications* 4 (2016), 39–46. https://doi.org/10.1016/j.vehcom.2016.04.001

[12] Michael Behrisch, Laura Bieker, Jakob Erdmann, and Daniel Krajzewicz. 2011. Sumo–simulation of urban mobility. In *The Third International Conference on Advances in System Simulation (SIMUL 2011), Barcelona, Spain*, Vol. 42.

[13] Paolo Bellavista, Alessandro Zanni, and Michele Solimando. 2017. A migration-enhanced edge computing support for mobile devices in hostile environments. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2017 13th International*. 957–962. https://doi.org/10.1109/IWCMC.2017.7986415

[14] Yuanguo Bi, Guangjie Han, Chuan Lin, Qingxu Deng, Lei Guo, and Fuliang Li. 2018. Mobility Support for Fog Computing: An SDN Approach. *IEEE Communications Magazine* 56, 5 (2018), 53–59. https://doi.org/10.1109/MCOM.2018.1700908

[15] Luiz F. Bittencourt, Márcio Moraes Lopes, Ioan Petri, and Omer F Rana. 2015. Towards virtual machine migration in fog computing. In *P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2015 10th International Conference on*. 1–8. https://doi.org/10.1109/3PGCIC.2015.85

[16] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. 2012. Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing - MCC '12*. 13–16. https://doi.org/10.1145/2342509.2342513

[17] Dario Bruneo, Salvatore Distefano, Francesco Longo, and Giovanni Merlino. 2016. An iot testbed for the software defined city vision: The# smartme project. In *Smart Computing (SMARTCOMP), 2016 IEEE International Conference on*. 1–6. https://doi.org/10.1109/SMARTCOMP.2016.7501678

[18] Roberto Bruschi, Franco Davoli, Paolo Lago, and Jane Frances Pajo. 2018. Move with Me: Scalably Keeping Virtual Objects Close to Users on the Move. In *2018 IEEE International Conference on Communications (ICC)*. 1–6. https://doi.org/10.1109/ICC.2018.8422229

[19] Roberto Bruschi, Franco Davoli, Chiara Lombardo, and Paolo Lago. 2018. Personal Services Placement and Low-Latency Migration in Edge Computing Environments. In *2018 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*.

[20] Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, César A F De Rose, and Rajkumar Buyya. 2011. CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software: Practice and experience* 41, 1 (2011), 23–50. https://doi.org/10.1002/spe.995

[21] Lucas Chaufournier, Prateek Sharma, Franck Le, Erich Nahum, Prashant Shenoy, and Don Towsley. 2017. Fast transparent virtual machine migration in distributed edge clouds. In *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*. 10. https://doi.org/10.1145/3132211.3134445

[22] Min Chen, Wei Li, Giancarlo Fortino, Yixue Hao, Long Hu, and Iztok Humar. 2019. A Dynamic Service Migration Mechanism in Edge Cognitive Computing. *ACM Transactions on Internet Technology (TOIT)* 19, 2 (2019), 30. https://doi.org/10.1145/3239565

[23] Mahfuzulhoq Chowdhury, Eckehard Steinbach, Wolfgang Kellerer, and Martin Maier. 2018. Context-Aware Task Migration for HART-Centric Collaboration over FiWi Based Tactile Internet Infrastructures. *IEEE Transactions on Parallel and Distributed Systems* 29, 6 (2018), 1231–1246. https://doi.org/10.1109/TPDS.2018.2791406

[24] Ertugrul N. Ciftcioglu, Kevin S Chan, Rahul Urgaonkar, Shiqiang Wang, and Ting He. 2015. Security-aware service migration for tactical mobile micro-clouds. In *Military Communications Conference, MILCOM 2015-2015 IEEE*. 1058–1063. https://doi.org/10.1109/MILCOM.2015.7357585

[25] Christopher Clark, Keir Fraser, Steven Hand, Jacob Gorm Hansen, Eric Jul, Christian Limpach, Ian Pratt, and Andrew Warfield. 2005. Live migration of virtual machines. In *Proceedings of the 2nd Conference on Symposium on Networked Systems Design & Implementation-Volume 2*. 273–286.

[26] CRIU [n. d.]. CRIU. Retrieved November 27, 2018 from https://www.criu.org/Main_Page

[27] Labhesh Deshpande and Kaikai Liu. 2017. Edge computing embedded platform with container migration. In *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*. 1–6. https://doi.org/10.1109/UIC-ATC.2017.8397578

[28] dpdk [n. d.]. The Intel Data-Plane Development Kit. Retrieved November 27, 2018 from http://www.dpdk.org

[29] Corentin Dupont, Raffaele Giaffreda, and Luca Capra. 2017. Edge computing in IoT context: Horizontal and vertical Linux container migration. In *2017 Global Internet of Things Summit (GIoTS)*. IEEE, 1–4. https://doi.org/10.1109/GIOTS.2017.8016218

[30] Wijaya Ekanayake, Heli Amarasinghe, and Ahmed Karmouch. 2017. SDN-based IaaS for mobile computing. In *Consumer Communications & Networking Conference (CCNC), 2017 14th IEEE Annual*. 179–184. https://doi.org/10.1109/CCNC.2017.7983102

[31] Qiang Fan, Nirwan Ansari, and Xiang Sun. 2017. Energy driven avatar migration in green cloudlet networks. *IEEE Communications Letters* 21, 7 (2017), 1601–1604. https://doi.org/10.1109/LCOMM.2017.2684812

[32] Ivan Farris, Tarik Taleb, Hannu Flinck, and Antonio Iera. 2018. Providing ultra-short latency to user-centric 5G applications at the mobile network edge. *Transactions on Emerging Telecommunications Technologies* 29, 4 (2018), e3169. https://doi.org/10.1002/ett.3169

[33] Sonja Filiposka, Anastas Mishev, and Katja Gilly. 2018. Community-based allocation and migration strategies for fog computing. In *Wireless Communications and Networking Conference (WCNC), 2018 IEEE*. 1–6. https://doi.org/10.1109/WCNC.2018.8377095

[34] Ramon R. Fontes, Samira Afzal, Samuel H.B. Brito, Mateus A.S. Santos, and Christian Esteve Rothenberg. 2015. Mininet-WiFi: Emulating software-defined wireless networks. In *Proceedings of the 11th International Conference on Network and Service Management, CNSM 2015*. 384–389. https://doi.org/10.1109/CNSM.2015.7367387

[35] Pantelis A. Frangoudis and Adlen Ksentini. 2018. Service migration versus service replication in Multi-access Edge Computing. In *2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*. 124–129. https://doi.org/10.1109/IWCMC.2018.8450284

[36] Lazaros Gkatzikis and Iordanis Koutsopoulos. 2013. Migrate or not? Exploiting dynamic task migration in mobile cloud computing systems. *IEEE Wireless Communications* 20, 3 (2013), 24–32. https://doi.org/10.1109/MWC.2013.6549280

[37] Diogo Gonçalves, Karima Velasquez, Marilia Curado, Luiz Bittencourt, and Edmundo Madeira. 2018. Proactive virtual machine migration in fog environments. In *2018 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 00742–00745. https://doi.org/10.1109/ISCC.2018.8538655

[38] Keerthana Govindaraj and Alexander Artemenko. 2018. Container Live Migration for Latency Critical Industrial Applications on Edge Computing. In *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, Vol. 1. 83–90. https://doi.org/10.1109/ETFA.2018.8502659

[39] Harshit Gupta, Amir Vahid Dastjerdi, Soumya K Ghosh, and Rajkumar Buyya. 2017. iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments. *Software: Practice and Experience* 47, 9 (2017), 1275–1296. https://doi.org/10.1002/spe.2509

[40] Kiryong Ha, Yoshihisa Abe, Zhuo Chen, Wenlu Hu, Brandon Amos, Padmanabhan Pillai, and Mahadev Satyanarayanan. 2015. Adaptive vm handoff across cloudlets. *Technical Report CMU-CS-15–113, CMU School of Computer Science* (2015).

[41] Kiryong Ha, Yoshihisa Abe, Thomas Eiszler, Zhuo Chen, Wenlu Hu, Brandon Amos, Rohit Upadhyaya, Padmanabhan Pillai, and Mahadev Satyanarayanan. 2017. You can teach elephants to dance: agile vm handoff for edge computing. In *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*. 12. https://doi.org/10.1145/3132211.3134453

[42] Kiryong Ha and Mahadev Satyanarayanan. 2015. Openstack++ for cloudlet deployment. *School of Computer Science Carnegie Mellon University Pittsburgh* (2015).

[43] Cheol-Ho Hong and Blesson Varghese. 2018. Resource Management in Fog/Edge Computing: A Survey. *arXiv preprint arXiv:1810.00305* (2018).

[44] Kirak Hong, David Lillethun, Umakishore Ramachandran, Beate Ottenwälder, and Boris Koldehofe. 2013. Opportunistic spatio-temporal event processing for mobile situation awareness. In *Proceedings of the 7th ACM international conference on Distributed event-based systems*. 195–206. https://doi.org/10.1145/2488222.2488266

[45] Motoshi Horii, Yuji Kojima, and Kenichi Fukuda. 2018. Stateful process migration for edge computing applications. In *Wireless Communications and Networking Conference (WCNC), 2018 IEEE*. 1–6. https://doi.org/10.1109/WCNC.2018.8377072

[46] Md. Mofijul Islam, Abdur Razzaque, and Jahidul Islam. 2016. A genetic algorithm for virtual machine migration in heterogeneous mobile cloud computing. In *Networking Systems and Security (NSysS), 2016 International Conference on*. 1–6. https://doi.org/10.1109/NSysS.2016.7400696

[47] Kai Kientopf, Saleem Raza, Simon Lansing, and Mesut Güneş. 2017. Service management platform to support service migrations for IoT smart city applications. In *Personal, Indoor, and Mobile Radio Communications (PIMRC), 2017 IEEE 28th Annual International Symposium on*. 1–5. https://doi.org/10.1109/PIMRC.2017.8292690

[48] Sunghwan Kim, Yohan Kim, and Hyuk Lim. 2016. Poster: Cyber-Function Virtualization for Mobile Cyber-Physical Systems. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services Companion*. 136. https://doi.org/10.1145/2938559.2938591

[49] Avi Kivity, Yaniv Kamay, Dor Laor, Uri Lublin, and Anthony Liguori. 2007. kvm: the Linux virtual machine monitor. In *Proceedings of the Linux symposium*, Vol. 1. 225–230.

[50] Adlen Ksentini, Tarik Taleb, and Min Chen. 2014. A Markov decision process-based service migration procedure for follow me cloud. In *Communications (ICC), 2014 IEEE International Conference on*. 1350–1354. https://doi.org/10.1109/ICC.2014.6883509

[51] Kubernetes [n. d.]. Production-Grade Container Orchestration. Retrieved December 17, 2018 from https://kubernetes.io/

[52] Chao Li, Yushu Xue, Jing Wang, Weigong Zhang, and Tao Li. 2018. Edge-Oriented Computing Paradigms: A Survey on Architecture Design and System Management. *ACM Computing Surveys (CSUR)* 51, 2 (2018), 39. https://doi.org/10.1145/3154815

[53] Márcio Moraes Lopes, Wilson A Higashino, Miriam A M Capretz, and Luiz Fernando Bittencourt. 2017. Myifogsim: A simulator for virtual machine migration in fog computing. In *Companion Proceedings of the10th International Conference on Utility and Cloud Computing*. 47–52. https://doi.org/10.1145/3147234.3148101

[54] Lele Ma, Shanhe Yi, Nancy Carter, and Qun Li. 2018. Efficient Live Migration of Edge Services Leveraging Container Layered Storage. *IEEE Transactions on Mobile Computing* (2018). https://doi.org/10.1109/TMC.2018.2871842

[55] Lele Ma, Shanhe Yi, and Qun Li. 2017. Efficient service handoff across edge servers via docker container migration. In *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*. 11. https://doi.org/10.1145/3132211.3134460

[56] Andrew Machen, Shiqiang Wang, Kin K Leung, Bong Jun Ko, and Theodoros Salonidis. 2016. Migrating running applications across mobile edge clouds: poster. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*. 435–436. https://doi.org/10.1145/2973750.2985265

[57] Andrew Machen, Shiqiang Wang, Kin K Leung, Bong Jun Ko, and Theodoros Salonidis. 2018. Live service migration in mobile edge clouds. *IEEE Wireless Communications* 25, 1 (2018), 140–147. https://doi.org/10.1109/MWC.2017.1700011

[58] Yuyi Mao, Changsheng You, Jun Zhang, Kaibin Huang, and Khaled B Letaief. 2017. A survey on mobile edge computing: The communication perspective. *IEEE Communications Surveys & Tutorials* 19, 4 (2017), 2322–2358. https://doi.org/10.1109/COMST.2017.2745201

[59] Eva Marín-Tordera, Xavi Masip-Bruin, Jordi García-Almiñana, Admela Jukan, Guang-Jie Ren, and Jiafeng Zhu. 2017. Do we all really know what a fog node is? Current trends towards an open definition. *Computer Communications* 109 (2017), 117–130. https://doi.org/10.1016/j.comcom.2017.05.013

[60] MECa [n. d.]. Multi-access Edge Computing (MEC). Retrieved November 27, 2018 from https://www.etsi.org/technologies-clusters/technologies/multi-access-edge-computing

[61] MECb [n. d.]. Mobile Edge Computing (MEC); End to End Mobility Aspects. Retrieved December 16, 2018 from https://www.etsi.org/deliver/etsi_gr/MEC/001_099/018/01.01.01_60/gr_MEC018v010101p.pdf

[62] Carla Mouradian, Diala Naboulsi, Sami Yangui, Roch H Glitho, Monique J Morrow, and Paul A Polakos. 2017. A comprehensive survey on fog computing: State-of-the-art and research challenges. *IEEE Communications Surveys & Tutorials* 20, 1 (2017), 416–464. https://doi.org/10.1109/COMST.2017.2771153

[63] MPTCP [n. d.]. MPTCP. Retrieved November 27, 2018 from https://www.multipath-tcp.org/

[64] Ahmad M. Mustafa, Omar M Abubakr, Omar Ahmadien, Ahmed Ahmedin, and Bassem Mokhtar. 2017. Mobility prediction for efficient resources management in vehicular cloud computing. In *Mobile Cloud Computing, Services, and Engineering (MobileCloud), 2017 5th IEEE International Conference on*. 53–59. https://doi.org/10.1109/MobileCloud.2017.24

[65] Apollinaire Nadembega, Abdelhakim Senhaji Hafid, and Ronald Brisebois. 2016. Mobility prediction model-based service migration procedure for follow me cloud to support QoS and QoE. In *Communications (ICC), 2016 IEEE International Conference on*. 1–6. https://doi.org/10.1109/ICC.2016.7511148

[66] Wahida Nasrin and Jiang Xie. 2018. SharedMEC: Sharing Clouds to Support User Mobility in Mobile Edge Computing. In *2018 IEEE International Conference on Communications (ICC)*. 1–6. https://doi.org/10.1109/ICC.2018.8422241

[67] OFC [n. d.]. OpenFog Consortium. Retrieved November 27, 2018 from https://www.openfogconsortium.org

[68] OpenStack [n. d.]. OpenStack. Retrieved December 17, 2018 from https://www.openstack.org/

[69] Opeyemi Osanaiye, Shuo Chen, Zheng Yan, Rongxing Lu, Kim-Kwang Raymond Choo, and Mqhele Dlodlo. 2017. From cloud to fog computing: A review and a conceptual live VM migration framework. *IEEE Access* 5 (2017), 8284–8300. https://doi.org/10.1109/ACCESS.2017.2692960

[70] Beate Ottenwälder, Boris Koldehofe, Kurt Rothermel, and Umakishore Ramachandran. 2013. MigCEP: operator migration for mobility driven distributed complex event processing. In *Proceedings of the 7th ACM international conference on Distributed event-based systems*. 183–194. https://doi.org/10.1145/2488222.2488265

[71] Beate Ottenwälder, Ruben Mayer, and Boris Koldehofe. 2014. Distributed complex event processing for mobile large-scale video applications. In *Proceedings of the Posters & Demos Session*. 5–6. https://doi.org/10.1145/2678508.2678511

[72] Tao Ouyang, Zhi Zhou, and Xu Chen. 2018. Follow Me at the Edge: Mobility-Aware Dynamic Service Placement for Mobile Edge Computing. *IEEE Journal on Selected Areas in Communications* (2018). https://doi.org/10.1109/IWQoS.2018.8624174

[73] phaul [n. d.]. P.Haul - CRIU. Retrieved November 27, 2018 from https://criu.org/P.Haul

[74] Carlo Puliafito, Enzo Mingozzi, and Giuseppe Anastasi. 2017. Fog computing for the internet of mobile things: issues and challenges. In *Smart Computing (SMARTCOMP), 2017 IEEE International Conference on*. 1–6. https://doi.org/10.1109/SMARTCOMP.2017.7947010

[75] Carlo Puliafito, Enzo Mingozzi, Carlo Vallati, Francesco Longo, and Giovanni Merlino. 2018. Companion Fog Computing: Supporting Things Mobility Through Container Migration at the Edge. In *2018 IEEE International*

Conference on Smart Computing (SMARTCOMP). 97–105. https://doi.org/10.1109/SMARTCOMP.2018.00079

[76] Carlo Puliafito, Enzo Mingozzi, Carlo Vallati, Francesco Longo, and Giovanni Merlino. 2018. Virtualization and migration at the network edge: an overview. In *2018 IEEE International Conference on Smart Computing (SMARTCOMP)*. 368–374. https://doi.org/10.1109/SMARTCOMP.2018.00031

[77] Tariq Qayyum, Asad Waqar Malik, Muazzam A Khan Khattak, Osman Khalid, and Samee U Khan. 2018. FogNetSim++: A toolkit for modeling and simulation of distributed fog environment. *IEEE Access* 6 (2018), 63570–63583. https://doi.org/10.1109/ACCESS.2018.2877696

[78] Yuqing Qiu, Chung-Horng Lung, Samuel Ajila, and Pradeep Srivastava. 2017. LXC container migration in cloudlets under multipath TCP. In *Computer Software and Applications Conference (COMPSAC), 2017 IEEE 41st Annual*, Vol. 2. 31–36. https://doi.org/10.1109/COMPSAC.2017.163

[79] Tarek K. Refaat, Burak Kantarci, and Hussein T Mouftah. 2016. Virtual machine migration and management for vehicular clouds. *Vehicular Communications* 4 (2016), 47–56. https://doi.org/10.1016/j.vehcom.2016.05.001

[80] Ioannis Sarrigiannis, Elli Kartsakli, Kostas Ramantas, Angelos Antonopoulos, and Christos Verikoukis. 2018. Application and Network VNF migration in a MEC-enabled 5G Architecture. In *2018 IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*. 1–6. https://doi.org/10.1109/CAMAD.2018.8514943

[81] Mahadev Satyanarayanan, Victor Bahl, Ramón Caceres, and Nigel Davies. 2009. The case for vm-based cloudlets in mobile computing. *IEEE pervasive Computing* (2009). https://doi.org/10.1109/MPRV.2009.82

[82] Enrique Saurez, Harshit Gupta, Ruben Mayer, and Umakishore Ramachandran. 2017. Demo abstract: Fog computing for improving user application interaction and context awareness. In *Internet-of-Things Design and Implementation (IoTDI), 2017 IEEE/ACM Second International Conference on*. 281–282. https://doi.org/10.1145/3054977.3057297

[83] Enrique Saurez, Kirak Hong, Dave Lillethun, Umakishore Ramachandran, and Beate Ottenwälder. 2016. Incremental deployment and migration of geo-distributed situation awareness applications in the fog. In *Proceedings of the 10th ACM International Conference on Distributed and Event-based Systems - DEBS '16*. 258–269. https://doi.org/10.1145/2933267.2933317

[84] Xiang Sun and Nirwan Ansari. 2016. EdgeIoT: Mobile edge computing for the Internet of Things. *IEEE Communications Magazine* 54, 12 (2016), 22–29. https://doi.org/10.1109/MCOM.2016.1600492CM

[85] Xiang Sun and Nirwan Ansari. 2016. PRIMAL: Profit maximization avatar placement for mobile edge computing. In *Communications (ICC), 2016 IEEE International Conference on*. 1–6. https://doi.org/10.1109/ICC.2016.7511131

[86] Tarik Taleb, Sunny Dutta, Adlen Ksentini, Muddesar Iqbal, and Hannu Flinck. 2017. Mobile edge computing potential in making cities smarter. *IEEE Communications Magazine* 55, 3 (2017), 38–43. https://doi.org/10.1109/MCOM.2017.1600249CM

[87] Tarik Taleb and Adlen Ksentini. 2013. An analytical model for follow me cloud. In *Global Communications Conference (GLOBECOM), 2013 IEEE*. 1291–1296. https://doi.org/10.1109/GLOCOM.2013.6831252

[88] Tarik Taleb and Adlen Ksentini. 2013. Follow me cloud: interworking federated clouds and distributed mobile networks. *IEEE Network* 27, 5 (2013), 12–19. https://doi.org/10.1109/MNET.2013.6616110

[89] Tarik Taleb, Adlen Ksentini, and Pantelis Frangoudis. 2016. Follow-me cloud: When cloud services follow mobile users. *IEEE Transactions on Cloud Computing* (2016). https://doi.org/10.1109/TCC.2016.2525987

[90] Tarik Taleb, Konstantinos Samdanis, Badr Mada, Hannu Flinck, Sunny Dutta, and Dario Sabella. 2017. On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration. *IEEE Communications Surveys & Tutorials* 19, 3 (2017), 1657–1681. https://doi.org/10.1109/COMST.2017.2705720

[91] Zhiqing Tang, Xiaojie Zhou, Fuming Zhang, Weijia Jia, and Wei Zhao. 2018. Migration Modeling and Learning Algorithms for Containers in Fog Computing. *IEEE Transactions on Services Computing* (2018). https://doi.org/10.1109/TSC.2018.2827070

[92] Fikirte Teka, Chung-Horng Lung, and Samuel Ajila. 2015. Seamless live virtual machine migration with cloudlets and multipath TCP. In *Computer Software and Applications Conference (COMPSAC), 2015 IEEE 39th Annual*, Vol. 2. 607–616. https://doi.org/10.1109/COMPSAC.2015.31

[93] Klervie Toczé and Simin Nadjm-Tehrani. 2018. A Taxonomy for Management and Optimization of Multiple Resources in Edge Computing. *Wireless Communications and Mobile Computing* 2018 (2018). https://doi.org/10.1155/2018/7476201

[94] Rahul Urgaonkar, Shiqiang Wang, Ting He, Murtaza Zafer, Kevin Chan, and Kin K Leung. 2015. Dynamic service migration and workload scheduling in edge-clouds. *Performance Evaluation* 91 (2015), 205–228. https://doi.org/10.1016/j.peva.2015.06.013

[95] Karima Velasquez, David Perez Abreu, Marilia Curado, and Edmundo Monteiro. 2017. Service placement for latency reduction in the internet of things. *Annals of Telecommunications* 72, 1-2 (2017), 105–115. https://doi.org/10.1007/s12243-016-0524-9

[96] Haoxin Wang, Jiang Xie, and Tao Han. 2018. A Smart Service Rebuilding Scheme across Cloudlets via Mobile AR Frame Feature Mapping. In *2018 IEEE International Conference on Communications (ICC)*. 1–6. https://doi.org/10.

1109/ICC.2018.8422226

[97] Shiqiang Wang, Rahul Urgaonkar, Ting He, Murtaza Zafer, Kevin Chan, and Kin K Leung. 2014. Mobility-induced service migration in mobile micro-clouds. In *Military Communications Conference (MILCOM), 2014 IEEE*. 835–840. https://doi.org/10.1109/MILCOM.2014.145

[98] Shiqiang Wang, Rahul Urgaonkar, Murtaza Zafer, Ting He, Kevin Chan, and Kin K Leung. 2015. Dynamic service migration in mobile edge-clouds. In *IFIP Networking Conference (IFIP Networking), 2015*. 1–9. https://doi.org/10.1109/IFIPNetworking.2015.7145316

[99] Shangguang Wang, Jinliang Xu, Ning Zhang, and Yujiong Liu. 2018. A Survey on Service Migration in Mobile Edge Computing. *IEEE Access* 6 (2018), 23511–23528. https://doi.org/10.1109/ACCESS.2018.2828102

[100] Zhe Wang, Zhangdui Zhong, Dongmei Zhao, and Minming Ni. 2018. Vehicle-Based Cloudlet Relaying for Mobile Computation Offloading. *IEEE Transactions on Vehicular Technology* 67, 11 (2018), 11181–11191. https://doi.org/10.1109/TVT.2018.2870392

[101] Hong Yao, Changmin Bai, Deze Zeng, Qingzhong Liang, and Yuanyuan Fan. 2015. Migrate or not? Exploring virtual machine migration in roadside cloudlet-based vehicular cloud. *Concurrency and Computation: Practice and Experience* 27, 18 (2015), 5780–5792. https://doi.org/10.1002/cpe.3642

[102] Shanhe Yi, Zijiang Hao, Zhengrui Qin, and Qun Li. 2015. Fog computing: Platform and applications. In *2015 Third IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb)*. 73–78. https://doi.org/10.1109/HotWeb.2015.22

[103] Ashkan Yousefpour, Caleb Fung, Tam Nguyen, Krishna Kadiyala, Fatemeh Jalali, Amirreza Niakanlahiji, Jian Kong, and Jason P Jue. 2019. All one needs to know about fog computing and related edge computing paradigms: a complete survey. *Journal of Systems Architecture* (2019). https://doi.org/10.1016/j.sysarc.2019.02.009

[104] Rong Yu, Yan Zhang, Stein Gjessing, Wenlong Xia, and Kun Yang. 2013. Toward cloud-based vehicular networks with efficient resource management. *IEEE Network* 27, 5 (2013), 48–55. https://doi.org/10.1109/MNET.2013.6616115

[105] Fei Zhang, Guangming Liu, Xiaoming Fu, and Ramin Yahyapour. 2018. A Survey on Virtual Machine Migration: Challenges, Techniques, and Open Issues. *IEEE Communications Surveys & Tutorials* 20, 2 (2018), 1206–1243. https://doi.org/10.1109/COMST.2018.2794881

[106] Wuyang Zhang, Jiachen Chen, Yanyong Zhang, and Dipankar Raychaudhuri. 2017. Towards efficient edge cloud augmentation for virtual reality MMOGs. In *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*. 8. https://doi.org/10.1145/3132211.3134463

[107] Wuyang Zhang, Yi Hu, Yanyong Zhang, and Dipankar Raychaudhuri. 2016. Segue: Quality of service aware edge cloud service migration. In *Cloud Computing Technology and Science (CloudCom), 2016 IEEE International Conference on*. 344–351. https://doi.org/10.1109/CloudCom.2016.0061

[108] Yan Zhang, Huimin Wu, Periklis Chatzimisios, and Shengli Xie. 2013. Virtual machine live migration for pervasive services in cloud-assisted vehicular networks. In *2013 8th International Conference on Communications and Networking in China (CHINACOM)*. 540–545. https://doi.org/10.1109/ChinaCom.2013.6694654

[109] Dan Zhao, Tan Yang, Yuehui Jin, and Yue Xu. 2017. A service migration strategy based on multiple attribute decision in mobile edge computing. In *Communication Technology (ICCT), 2017 IEEE 17th International Conference on*. 986–990. https://doi.org/10.1109/ICCT.2017.8359782

[110] Peiyue Zhao and György Dán. 2018. Time Constrained Service-Aware Migration of Virtualized Services for Mobile Edge Computing. In *2018 30th International Teletraffic Congress (ITC 30)*, Vol. 01. 64–72. https://doi.org/10.1109/ITC30.2018.00017

[111] Bowen Zhou and Rajkumar Buyya. 2018. Augmentation techniques for mobile cloud computing: A taxonomy, survey, and future directions. *ACM Computing Surveys (CSUR)* 51, 1 (2018), 13. https://doi.org/10.1145/3152397

[112] Chao Zhu, Giancarlo Pastor, Yu Xiao, Yong Li, and Antti Ylae-Jaeaeski. 2018. Fog Following Me: Latency and Quality Balanced Task Allocation in Vehicular Fog Computing. In *2018 15th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. 1–9. https://doi.org/10.1109/SAHCN.2018.8397129