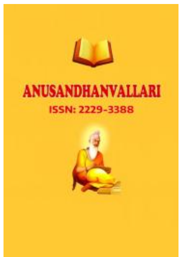

Anusandhanvallari

Vol 2023, No. 01

December 2023

ISSN 2229-3388

Mobility-Aware Service Migration in Fog: A Comprehensive Literature Review

Saravjit Chahal[1], Anita Singhrova[2]

[1,2]Department of Computer Science and Engineering,

Deenbandhu Chhotu Ram University of Science and Technology, Murthal, Haryana, India

Email: saravjitchahal.schcse@dcrustm.org

Abstract

Mobility-induced service migration has become pivotal in fog computing to maintain low latency and service continuity for moving users and devices. This literature review provides an in-depth analysis of primarily peer-reviewed journal research on mobility-aware service placement and migration techniques in fog environments. We first discuss the scope and motivation, highlighting the challenges that frequent mobility poses to edge-hosted applications. We present a thematic synthesis that categorizes the state-of-the-art into optimization-based frameworks (e.g., cost/reward-driven and heuristic algorithms), machine learning and reinforcement learning approaches, proactive vs. reactive handoff strategies, and system-level solutions (architectures, platforms, simulators) enabling live migration. Two comparison tables summarize the key approaches and the platforms/tools for mobility support. A critical analysis is provided, contrasting techniques in terms of decision models, overheads, QoS outcomes, and applicability across scenarios like smart cities, vehicular networks, and industry 4.0. We identify contradictions and trade-offs – for example, between minimizing latency and limiting migration frequency or energy use – reported across studies. Furthermore, research gaps are delineated, including the need for more accurate mobility prediction, energy-efficient yet low-latency migrations, and improved security in multi-stakeholder (federated) fog environments. Finally, future directions are suggested, pointing toward holistic mobility management frameworks, including emerging AI-enabled approaches for adaptive migration, and real-world testbed validations to bridge the gap between simulation and deployment. This review aims to guide researchers in developing robust fog systems that gracefully handle user mobility while meeting stringent Quality of Service requirements.

Keywords: Fog computing; Service migration; Mobility management; Quality of Service (QoS); Internet of Things (IoT).

Introduction

Internet of Things (IoT) applications are increasingly deployed on fog computing infrastructure to meet stringent latency and bandwidth requirements that distant cloud data centers cannot satisfy [1]. By placing computational services on fog nodes closer to end-users, network delays can be significantly reduced, enabling real-time processing for applications like augmented reality, connected vehicles, and smart cities [2]. However, a major challenge emerges when users or devices are mobile: as they move out of one fog node's coverage and into another's, the service instance serving them must migrate or be re-instantiated on a closer node to maintain low end-to-end latency and avoid service disruption [3]. This process is often referred to as mobility-induced service migration, or colloquially, a "follow-me" service paradigm where the compute follows the user. In this review, the terms fog computing, edge computing, and mobile edge computing (MEC) are used interchangeably when referring to compute resources deployed at the network edge to support low-latency services, unless a distinction is explicitly required by the cited work. The concept of migrating services to follow mobile users was introduced as early as 2013 in the context of cloud computing. Taleb *et al.* proposed the Follow-Me Cloud framework, wherein cloud services could dynamically relocate between geographically distributed data centers to maintain proximity to moving users [4]. This idea was implemented using Software-Defined Networking (SDN) to redirect users' traffic to the new service location, thereby achieving *seamless handover* at the network level [5]. While Follow-Me Cloud laid the groundwork for mobility-aware service continuity, it primarily considered migrations across cloud data centers. The emergence of fog and edge computing brought this paradigm to the network edge, where service instances run on micro data centers or cloudlets attached to access points (e.g., base stations or roadside units). Here, mobility is even more frequent and granular (e.g., a user's smartphone or a connected vehicle moving between adjacent road side fog units), making efficient service migration in fog environments a critical research problem.

In fog computing scenarios, the goal is to transfer the user's application state or service container/VM from the current fog node to a target fog node with minimal latency, minimal packet loss, and no noticeable downtime in the service [6]. Achieving this is non-trivial due to constrained resources at the edge, the potential frequency of handovers, and the overhead of migration itself (state transfer time, buffering, etc.) [7]. Researchers have approached this challenge from multiple angles. Some works focus on when and where to migrate – optimizing the service placement dynamically to minimize latency or cost given user mobility patterns [8]. Others concentrate on how to migrate efficiently – for example, using lightweight containerization and live migration techniques to reduce downtime [9]. Several illustrative scenarios underscore the importance of mobility-aware service migration. In smart vehicular networks, vehicles offload tasks (like video analysis or path planning) to roadside fog nodes; as a car drives, the offloaded service must migrate to keep latency low. This issue was addressed with Follow Me Fog (FMF), a framework for *seamless handover timing* in fog environments [6]. By monitoring signal strength and performing *job pre-migration* slightly *before* the vehicle switches base stations, their prototype achieved 36% lower latency during handovers. In smart city IoT, a user carrying a smart health device might move across different Wi-Fi or 5G cells; the personal monitoring service should relocate accordingly. Ouyang *et al.* tackled such scenarios by optimizing a cost-performance trade-off: they formulated an online problem to minimize long-term latency under an energy/cost budget, designing a distributed solution that decides migration on-the-fly without needing exact future mobility information [1].

This review is motivated by the growing body of research addressing these challenges and the need to consolidate findings across disparate approaches (optimization models, algorithmic heuristics, system implementations, etc.). This review focuses mainly on peer-reviewed journal literature relevant to mobility-aware service migration in fog computing. By reviewing these works, we aim to (1) provide a structured overview of current strategies for mobility-aware service migration in fog computing, (2) compare their effectiveness and assumptions, and (3) identify open issues that require further study. Ultimately, enabling robust mobility management in fog computing will unlock truly ubiquitous computing experiences, where users moving through space continue to receive responsive digital services without interruption. The remainder of this paper is organized as follows. The Scope and Organization section then outlines the focus and structure of the review. In Thematic Synthesis, we present taxonomy of approaches, grouping the literature into thematic categories. A Comparative Analysis is provided through summary tables and discussion, highlighting differences in objectives (latency, energy, cost), solution techniques, and evaluation settings. We then offer a Critical Discussion of the strengths and limitations observed, and identify Gaps & Open Issues supported by evidence from the literature. In Future Directions, we suggest promising research avenues building on current progress. Finally, the Conclusion summarizes key insights and takeaways from this review.

Scope and Organization of the Review

This review focuses on peer-reviewed journal studies that investigate mobility-aware service migration and dynamic service placement in fog and edge computing environments. Rather than aiming for an exhaustive or systematic survey, the review emphasizes representative and influential works that illustrate key design philosophies, algorithmic strategies, and system implementations proposed in the literature. The selected studies are synthesized thematically to highlight common trends, contrasting assumptions, and reported trade-offs in terms of latency, migration overhead, energy consumption, and Quality of Service. This thematic organization provides the basis for the comparative analysis and critical discussion presented in the subsequent sections. In the next section, we will comparatively analyze representative works from these themes, using tables to line up their key characteristics and results, and we will discuss how they complement or contradict each other. This sets the stage for a critical discussion of strengths and weaknesses of the approaches, as well as identification of open issues.

Comparative Analysis

To compare the diverse approaches identified in the thematic synthesis, we structure our analysis along two dimensions: (A) the migration decision strategies (optimization vs. heuristic vs. learning, reactive vs. proactive), and (B) the system implementation aspects (how migrations are executed and supported by frameworks). We use two comparative tables to summarize key differences, followed by interpretative discussion. Table 1 compares a selection of approaches on decision strategy, input requirements, and performance outcomes.

Table 1. Comparison of migration decision strategies.

Approach (Year)	**Decision Strategy**	**Optimization Objective**	**Reported Performance**
Ouyang *et al.* (2018) [1]	Online optimization (Lyapunov-based) – Reactive (no prediction)	Minimize long-term *latency* subject to migration *cost* constraint	Achieved optimal latency with 50% less cost.
Wang *et al.* (2017) [10]	Markov Decision Process – solved via Threshold Policy (theoretical optimal)	Minimize expected sum of latency + migration penalty (long-run average)	Outperformed reactive baseline by 30% lower cost.
Martin *et al.* (2020) [11]	Hybrid Autonomic: Rule-based trigger + GA heuristic (MAPE loop) – Proactive	Minimize combined network usage, delay, and cost (multi-objective in GA fitness)	20–25% less network traffic and 15% lower avg. response time.
Aleyadeh *et al.* (2022) [12]	Integer Programming (optimal off-line) + heuristic (EC2-MRI) – Reactive	Minimize service downtime while meeting latency requirements (choose migrate vs. reinstantiate optimally)	Optimal model (OC-MRI) reduced downtime by 40% compared to always-migrate and always-restart.

H. Wang *et al.* (2023) [13]	Deep Reinforcement Learning (DQN) – Reactive learning (no explicit prediction, but learns policy)	Minimize weighted sum of latency + migration cost per episode (modeled via reward)	Achieved 18% lower cost than a threshold heuristic and 30% lower than never-migrate.
Bozkaya (2023) [3]	Predict-and-optimize (Digital Twin + solver) – Proactive	Minimize task completion time (latency) given predicted mobility and task deadlines	Yielded 10–15% shorter completion times in scenario tests. Fewer migrations needed.

Most approaches in Table 1 report clear performance gains compared to baseline strategies such as ‘never migrate,’ ‘always migrate,’ or naive static schemes, indicating the importance of informed migration decisions. These numbers, while promising, often come from simulation with particular assumptions. For instance, Ouyang’s optimal latency is theoretical – which is a strong result given it doesn’t use prediction. RL’s 18% cost reduction depends on how cost is weighted. It’s hard to directly compare these percentages because the baselines differ.

Subsequently, Table 2 compares system- and implementation-oriented aspects, including the virtualization medium (VMs or containers), migration mechanisms for minimizing downtime, and supporting technologies such as SDN or specific orchestration frameworks, thereby highlighting practical feasibility and system complexity.

Table 2. Comparison of system implementation and migration execution aspects.

Work & Year	**Migration Medium** (VM/Container/Function)	**Mechanism** (Live migration, Re-instantiation, etc.)	**Downtime / QoS Handling**
Bao *et al.* (2017) – FMF [6]	Container (computation job in container)	Live pre-copy migration (job state pre-migrated before handover trigger)	Achieved *seamless* handover: measured downtime negligible (~0.2s) and 36.5% latency reduction during handover.
Puliafito *et al.* (2019) – Perf Eval [14]	Container (Docker)	Tested: Cold vs. Live (pre-copy) vs. Post-copy vs. Hybrid	Recommended live for latency-critical cases, hybrid if network unstable.
Martin *et al.* (2020) – MAMF [11]	Container (Docker within iFogSim simulation)	Modeled as stop-start migration (with some delay to transfer state) in simulator	If migration overhead would violate SLA, it opts not to migrate. Ensures deadline QoS met ~98% of time.
Okwuibe *et al.* (2020) – IIoT Orchestration [15]	Container	Combination: if possible, keeps container running and just re-routes flows	By using SDN, they eliminate TCP session breakage – network paths reconfigured on the fly to new edge node.
Puliafito *et al.* (2021) –	VM (used OpenStack VMs)	Live migration via OpenStack Nova (pre-copy	Observed average service interruption <10 ms – effectively

OpenStack Fog [16]		for VMs) or Docker live migration for containers	hitting 10 ms one-way latency target.
Aleyadeh *et al.* (2022) – OC-MRI [13]	Container	Both: either *Live migrate* container or *Instantiate new* instance from image at target	Optimal selection minimized downtime to <1s in cases where naive would be 2–3s.
iFogSim2 (2022) – Mahmud [17]	Supports VMs, containers, and modules (microservices) as simulation abstractions	Can simulate live migration or stop-restart depending on user choice	Allows modeling of downtime explicitly for each migration event.
Bozkaya (2023) – DT approach [3]	Implied by system model (task/ container abstraction)	Optimization can decide migrate vs. keep	Ensures "acceptable QoS" by classification of tasks: delay-sensitive tasks get priority in migration decisions.
H. Wang (2023) – SMDQN [12]	Not explicitly defined	RL policy decides migrate or not at each time slot;	Better stability of service cost, implying fewer disruptive migrations.

Table 2 highlights clear implementation trends in mobility-aware migration solutions. Recent works predominantly adopt container-based virtualization due to its lower overhead compared to VMs, which is critical for edge environments. Live (pre-copy) migration is widely used to minimize downtime, while hybrid or re-instantiation strategies are preferred when network conditions or state size make live migration costly. Several studies emphasize the role of SDN or centralized controllers in maintaining service continuity through dynamic traffic redirection. Overall, the comparison shows that effective QoS preservation depends on a careful combination of lightweight virtualization, appropriate migration mechanisms, and network-level support, with trade-offs between latency, energy consumption, and system complexity.

In summary, our comparative analysis shows that a variety of strategies can be effective, each with its assumptions: threshold policies work well under well-modeled scenarios, learning can adapt to unknown scenarios given training, proactive methods excel with accurate prediction, and robust system support (like SDN and container tech) is crucial to realize any of these strategies in practice with minimal disruption. Next, we move to a critical discussion of these findings, examining strengths, limitations, and how different approaches complement or conflict with each other.

The evaluation shows that the approaches reviewed are largely complementary, each addressing slices of the problem (theoretical vs. practical, reactive vs. proactive). There is a convergence that smart migration policies can greatly enhance QoS as compared to static edge deployment. However, no single approach solves all problems: e.g., one might do great on latency but not consider energy, another handles energy but ignores security, etc. Implementation feasibility is demonstrated in small scale, but scalability and integration into real networks remain open. There are some underexplored tensions (e.g., security vs. performance, multi-user fairness, standardizing evaluation) that form the basis of open issues.

Next, we articulate these open issues and research gaps more formally, backed by evidence from our review, and suggest future directions to address them.

Gaps & Open Issues

Despite substantial progress in mobility-aware fog service migration, our review has uncovered numerous open challenges and research gaps that remain to be addressed. We highlight key gaps, each supported by insights from the literature:

1. **Accurate Mobility Prediction:** While some approaches leverage mobility prediction (e.g., Bozkaya's DT with HMM [3]), predicting user movement reliably is still an open problem. Mobility can be stochastic and context-dependent. Mis-predictions can trigger unnecessary migrations, incurring overhead with no benefit. Future research must improve prediction methods (perhaps via machine learning on rich contextual data) and develop migration policies robust to prediction errors (e.g., rollback mechanisms if a predicted move doesn't occur).
2. **Multi-Metric Optimization (Latency–Energy Trade-off):** Most studies optimize for latency or cost individually, but fog computing often demands *joint* optimization. For instance, Fan *et al.* (2017) prioritized energy savings in green cloudlets [7] whereas Ouyang (2018) prioritized latency under a cost budget [1]. There is a gap in frameworks that can flexibly balance multiple QoS metrics (latency, energy, bandwidth, etc.) simultaneously.
3. **Scalability to Many Devices and Services:** Many algorithms have been tested with a handful of nodes or single service scenarios. It remains unclear how they perform with hundreds of moving users and dozens of services concurrently. The complexity of solving placement for many services can grow combinatorially (an NP-hard problem akin to generalized assignment) [8].
4. **Standardized Evaluation and Benchmarks:** As noted in the discussion, there's a lack of common benchmarks. Each paper uses its own topology, mobility model, and workload pattern, making it difficult to directly compare approaches. A gap exists for establishing standard mobility traces and edge network scenarios (similar to how the networking community uses standard workloads or topologies for routing protocols).
5. **Security and Privacy in Migration:** Most reviewed works do not address security. However, migrating services involves moving code and state (which may include sensitive data) across infrastructure. This gap is increasingly critical as fog nodes could be operated by different providers, and users need assurance their service handoff doesn't expose them to risk.

Future Directions

Based on the identified research gaps, several key directions can guide future work in mobility-aware fog/edge service migration:

1. **AI-Driven and Learning-Based Orchestration:** Integrating machine learning, particularly reinforcement learning and deep learning, can enable adaptive and predictive migration decisions. Learning-based models can optimize long-term QoS–cost trade-offs, improve mobility prediction, and dynamically tune migration policies. Federated learning further offers decentralized training while preserving privacy, though real-time responsiveness and robustness remain critical challenges.
2. **Collaborative and Multi-Domain Edge Migration:** Future systems should support cooperative migration among multiple edge nodes and administrative domains. Information sharing, distributed coordination, and

standardized container-based mechanisms can enable load balancing, group mobility handling, and seamless inter-edge handovers within federated or "edge mesh" architectures.

3. **Hybrid Migration–Replication Strategies:** Combining proactive service replication with migration can reduce downtime by pre-instantiating services at predicted locations. Research is needed to determine when replication is preferable to migration, considering factors such as state size, resource overhead, and user mobility patterns.
4. **Energy-Aware and Green Migration:** Sustainability-focused migration strategies should account for energy consumption, renewable availability, and energy pricing. Incorporating energy cost models into migration decisions can enable greener orchestration without compromising performance.
5. **Advanced Simulation and Benchmarking:** More realistic simulators and shared benchmarks incorporating real mobility traces, network dynamics, and user behavior are needed for fair evaluation. Open-source testbeds and emulation platforms can further bridge the gap between simulation and deployment.

Overall, future mobility-aware edge migration research is expected to evolve toward intelligent, cooperative, secure, energy-efficient, and practically deployable systems that support seamless mobile experiences under real-world constraints.

Conclusion

Mobility-aware service migration is a core capability of fog and edge computing, enabling low-latency and reliable services for mobile users. This review analyzed key journal works and showed that migration is practically feasible using techniques such as live migration, predictive handoff, and optimized re-instantiation. The study highlights that no single approach is optimal for all scenarios; effective solutions balance latency, overhead, and energy while adapting to mobility context and workload characteristics. Proactive, reactive, and multi-objective strategies each offer distinct advantages, yet challenges remain in scalability, security, benchmarking, and multi-domain interoperability. In summary, mobility-aware service migration is a rapidly maturing field and a foundation for future edge-enabled applications. Addressing the remaining gaps through intelligent, secure, and energy-aware designs aligned with 5G/6G ecosystems will be key to delivering truly seamless, user-centric edge services.

References

[1] T. Ouyang, Z. Zhou, and X. Chen, "Follow me at the edge: Mobility-aware dynamic service placement for mobile edge computing," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 10, pp. 2333–2345, 2018. https://doi.org/10.1109/JSAC.2018.2869954

[2] R. Basir, S. Qaisar, M. Ali, M. Aldwairi, M. I. Ashraf, A. Mahmood, and M. Gidlund, "Fog computing enabling industrial Internet of Things: State-of-the-art and research challenges," *Sensors*, vol. 19, art. no. 4807, 2019. https://doi.org/10.3390/s19214807

[3] E. Bozkaya, "Digital twin-assisted and mobility-aware service migration in mobile edge computing," *Computer Networks*, vol. 231, art. no. 109798, Jul. 2023. https://doi.org/10.1016/j.comnet.2023.109798

[4] T. Taleb and A. Ksentini, "Follow-me cloud: Interworking federated clouds and distributed mobile networks," *IEEE Network*, vol. 27, no. 5, pp. 12–19, 2013. https://doi.org/10.1109/MNET.2013.6616110

[5] Y. S. Chen and Y.-T. Tsai, “A mobility management using follow-me cloud-cloudlet in fog-computing-based RANs for smart cities,” *Sensors*, vol. 18, no. 2, art. no. 489, Feb. 2018. https://doi.org/10.3390/s18020489

[6] W. Bao, D. Yuan, Z. Yang, S. Wang, W. Li, and A. Y. Zomaya, “Follow me fog: Toward seamless handover timing schemes in a fog computing environment,” *IEEE Communications Magazine*, vol. 55, no. 11, pp. 72–78, 2017. https://doi.org/10.1109/MCOM.2017.1700363

[7] Q. Fan, N. Ansari, and X. Sun, “Energy driven avatar migration in green cloudlet networks,” *IEEE Communications Letters*, vol. 21, no. 7, pp. 1601–1604, 2017. https://doi.org/10.1109/LCOMM.2017.2684812

[8] D. Zhao, G. Sun, D. Liao, S. Xu, and V. Chang, “Mobile-aware service function chain migration in cloud–fog computing,” *Future Generation Computer Systems*, vol. 96, pp. 591–604, 2019. https://doi.org/10.1016/j.future.2019.02.031

[9] K. Govindaraj and A. Artemenko, “Container live migration for latency-critical industrial applications on edge computing,” in *Proceedings of the 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, Turin, Italy, 2018, pp. 83–90. https://doi.org/10.1109/ETFA.2018.8502659

[10] S. Wang, R. Urgaonkar, T. He, K. Chan, M. Zafer, and K. Leung, “Dynamic service placement for mobile micro-clouds with predicted future costs,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 4, pp. 1002–1016, 2017. https://doi.org/10.1109/TPDS.2016.2604814

[11] J. P. Martin, A. Kandasamy, and K. Chandrasekaran, “Mobility aware autonomic approach for the migration of application modules in fog computing environment,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 11, pp. 5259–5278, 2020. https://doi.org/10.1007/s12652-020-01854-x

[12] S. Aleyadeh, A. Moubayed, P. Heidari, and A. Shami, “Optimal container migration/re-instantiation in hybrid computing environments,” *IEEE Open Journal of the Communications Society*, vol. 3, pp. 15–30, 2022. https://doi.org/10.1109/OJCOMS.2022.3140272

[13] H. Wang, Y. Li, A. Zhou, Y. Guo, and S. Wang, “Service migration in mobile edge computing: A deep reinforcement learning approach,” *International Journal of Communication Systems*, vol. 36, no. 1, e4413, 2023. https://doi.org/10.1002/dac.4413

[14] C. Puliafito, C. Vallati, E. Mingozzi, G. Merlino, F. Longo, and A. Puliafito, “Container migration in the fog: A performance evaluation,” *Sensors*, vol. 19, no. 7, art. no. 1488, 2019. https://doi.org/10.3390/s19071488

[15] J. Okwuibe, J. Haavisto, E. Harjula, I. Ahmad, and M. Ylianttila, “SDN enhanced resource orchestration for industrial IoT in containerized edge applications,” *IEEE Access*, vol. 8, pp. 229117–229131, 2020. https://doi.org/10.1109/ACCESS.2020.3045563

[16] C. Puliafito, C. Vallati, E. Mingozzi, G. Merlino, and F. Longo, “Design and evaluation of a fog platform supporting device mobility through container migration,” *Pervasive and Mobile Computing*, vol. 74, art. no. 101415, 2021. https://doi.org/10.1016/j.pmcj.2021.101415

[17] R. Mahmud, S. Pallewatta, M. Goudarzi, and R. Buyya, “iFogSim2: An extended iFogSim simulator for mobility, clustering, and microservice management in edge and fog computing environments,” *Journal of Systems and Software*, vol. 190, art. no. 111351, 2022. https://doi.org/10.1016/j.jss.2022.111351