Edge Server Placement Considering Resilience in Mobile Edge Computing Networks

by

Syeda Mahfuza Begum
B.Sc., Bangladesh University of Engineering and Technology (BUET), 2006

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

in the Department of Computer Science

Edge Server Placement Considering Resilience in Mobile Edge Computing Networks

by

Syeda Mahfuza Begum
B.Sc., Bangladesh University of Engineering and Technology (BUET), 2006

Supervisory Committee

Dr. Jianping Pan, Supervisor
(Professor, Department of Computer Science)

Dr. Sudhakar Ganti, Departmental Member
(Associate Professor, Department of Computer Science)

ABSTRACT

In today's rapidly evolving communication landscape, the demand for exceptional Quality of Service (QoS) and Quality of Experience (QoE) in communication networks has reached unprecedented levels. This surge in demand can be attributed to the explosive growth and pervasive deployment of Internet infrastructure. Emerging technologies and novel applications underscore the urgency for a network architecture that not only delivers speed and efficiency but also boasts scalability and resilience beyond the capabilities of traditional cloud computing networks. Mobile Edge Computing (MEC) stands as a promising solution to address these challenges. By deploying Edge Servers (ESs) in close proximity to end-user devices, MEC enables the offloading of delay-sensitive and computationally intensive workloads from mobile applications. This deployment, in turn, mitigates latency issues and enhances the QoE for mobile users. However, the reliability of Edge Server Placement (ESP) within MEC networks is of paramount importance. While several studies have explored the ESP problem in MEC networks, they often focus on two main objectives: minimizing Edge Server (ES) access delay and optimizing workload distribution. However, one critical aspect has been relatively under-emphasized: the resiliency of ESP. The failure or malfunction of ESs, stemming from various challenges, can disrupt operations and degrade the overall QoS/QoE of the network. In this study, we tackle the ESP problem in MEC networks from a distinctive perspective. Our focal point is to minimize ES access delay, efficiently balance workloads, and significantly enhance network resilience. To achieve these objectives, our innovative algorithm employs a dual strategy. First, we utilize the robust K-medoids clustering algorithm for ESP, optimizing the architectural layout of MEC networks. Second, we introduce a bespoke heuristic algorithm designed to allocate multiple ESs to each Base Station (BS), thereby fortifying network resilience. This approach not only adheres to various constraints but also ensures uninterrupted services, even in the face of server failures, while consistently meeting key performance indicators. Experimental results, based on real-world data, prove the effectiveness of our algorithm. It not only reduces access delay and workload imbalances but also ensures responsive performance and uninterrupted services, even in scenarios involving ES failures.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AI	Artificial Intelligence
AR	Augmented Reality
BS	Base Station
CDN	Content Delivery Network
DRL	Deep Reinforcement Learning
ES	Edge Server
ESP	Edge Server Placement
FCS	Fog Computing Systems
FIRE	Failure-Adaptive Reinforcement Learning
FLP	Facility Location Problem
IIoT	Industrial Internet of Things
ILP	Integer Linear Programming
IoD	Internet of Drones
IoMT	Internet of Medical Things
IoT	Internet of Things
IoV	Internet of Vehicles
MEC	Mobile Edge Computing
MILP	Mixed-Integer Linear Programming
MIP	Mixed Integer Program
MIQP	Mixed Integer Quadratic Program
ML	Machine Learning
MR	Mixed Reality
PS-LTE	Public Safety Long-Term Evolution
QoE	Quality of Experience
QoS	Quality of Service
RECS	Resilient Programmable System for Edge Cloud Systems
RESP	Resilient Edge Server Placement
RkESP	Robustness-oriented k Edge Server Placement
SDN	Software-Defined Networking
SLA	Service Level Agreements
UAV	Unmanned Aerial Vehicles
UE	User Equipment
VANET	Vehicular Ad Hoc Networks
V2E	Vehicle-to-Edge
VM	Virtual Machine
VNF	Virtual Network Function
VR	Virtual Reality
WAN	Wide Area Network

ACKNOWLEDGEMENTS

I would like to express my heartfelt gratitude to the following:

- **My Family, Friends, and Even the Weather**: Thank you for standing by me during challenging times and providing unwavering support.
- **My Supervisor**: I am immensely thankful to Dr. Jianping Pan for his invaluable mentorship, support, encouragement, and boundless patience.
- **Supervisory Committee Member**: I am also thankful to Professor Sudhakar Ganti, for his invaluable assistance and support throughout this endeavor.
- **External Academic Mentors**:
  - Dr. Rukhsana Afroz Ruby: I express my deep appreciation for her guidance and support during the pursuit of my master's thesis.
  - Dr. Humayun Kabir: I am grateful for the valuable insights and assistance that significantly contributed to the completion of my master's thesis.
- **Academic Writing Expert at UVic**: I am grateful to Nancy Ami, Manager of the Centre for Academic Communication at the University of Victoria, for her invaluable assistance and guidance in improving the academic writing quality of the thesis.
- **NSERC and Cassels Shaw**: I extend my sincere appreciation for their generous scholarship, which made it possible for me to pursue my academic aspirations.
- **Lab Mates**: Lastly, I extend my sincere appreciation to all those who have been part of my master's program, especially my fellow lab mates Jinwei Zhao, Mostafa Abdollahi, and Zhiming Huang, for their valuable comments on my thesis.

"Success is not final, failure is not fatal: It is the courage to continue that counts."
- Winston Churchill

DEDICATION

This work is a heartfelt tribute to my family, especially my late parents and my dear children, Muaz and Mahnoor. It is also a warm embrace for my loving brothers and sisters. You all mean the world to me, and your love and support have lit up every step of my journey.

Chapter 1

Introduction

1.1 Mobile Edge Computing (MEC)

In today's fast-paced digital landscape, two significant trends have reshaped how we use technology: the widespread adoption of mobile devices and the rapid growth of the Internet of Things (IoT) [1]. Mobile phones, especially smartphones, have become an integral part of our lives, serving as tools for entertainment, communication, and work. Simultaneously, the rise of smart applications and services, powered by technologies like machine learning, has introduced new levels of intelligence and convenience. However, these sophisticated applications demand substantial computing resources.

The challenge is that most mobile devices have limited processing power, storage, and battery life. To overcome these limitations, one solution is to offload computationally intensive tasks to powerful remote cloud servers. However, an important issue arises: cloud servers are often far away from mobile devices, causing delays in accessing services. For applications where real-time responses matter, like online gaming or Augmented Reality (AR), these delays are problematic. In essence, traditional cloud computing does not provide the speed needed for responsive mobile applications.

This context is where MEC has an impact, fundamentally altering the game. Instead of relying on distant cloud servers, MEC brings computing power (edge server) closer to mobile devices. Edge Servers (ESs), strategically placed at the edge of the network, enable quicker processing of tasks, reducing delays significantly. In effect, it is akin to having a localized cloud infrastructure in immediate proximity, as illustrated in Figure 1.1.

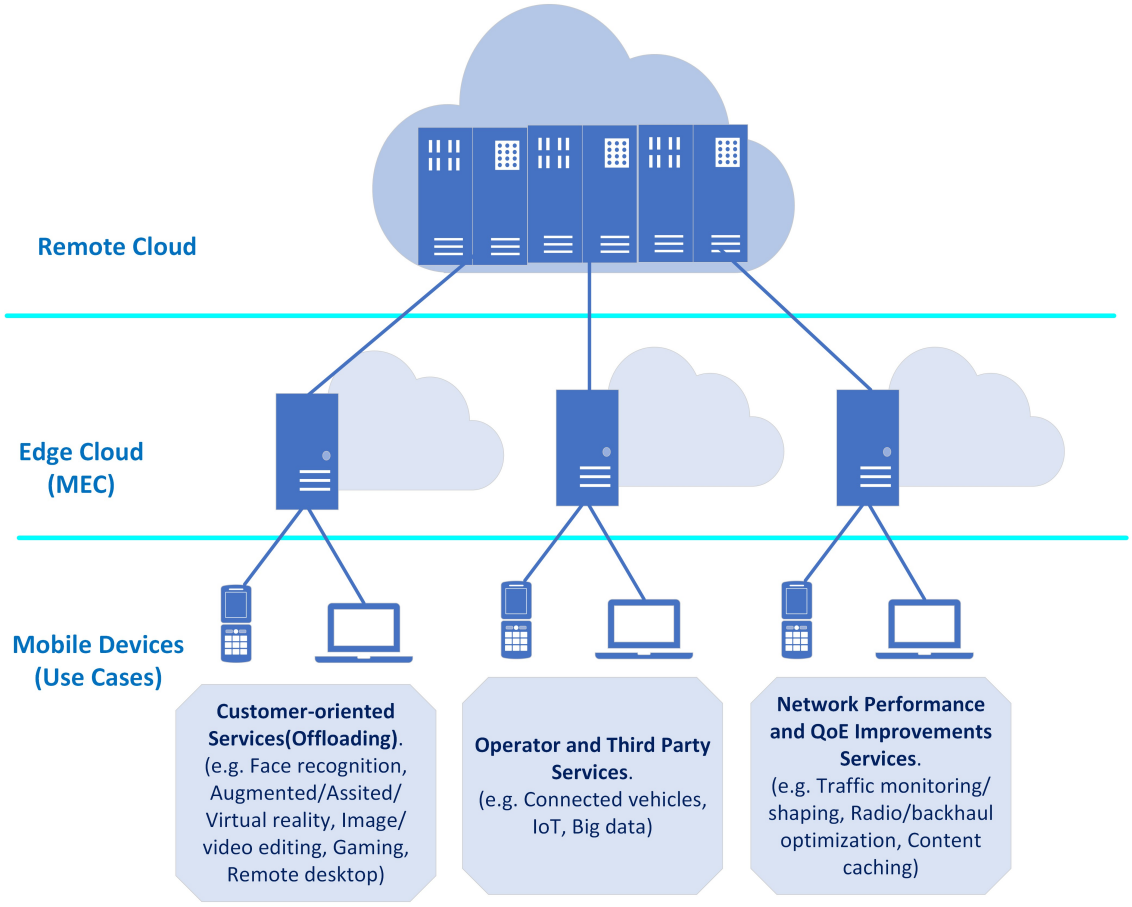


Figure 1.1: An architecture of MEC network.

Now, let us introduce an additional layer of resiliency into this scenario. Resiliency is crucial because communication networks can face various challenges, from technical glitches to natural disasters, and even malicious attacks [2]. These challenges can disrupt services and have serious consequences. To ensure that the systems can withstand these challenges and continue functioning, they need to be resilient. In the context of our thesis, resilience means ensuring that, even if an ES fails, the system can adapt and keep providing services without significant interruptions. Think of it as having a backup plan in case something goes wrong.

So, here is the core of our research: we are diving deep into the world of MEC to figure out where to place these ESs. This placement needs to achieve three critical goals. First, it should minimize delays for mobile users, ensuring they get quick access to services. Second, it should distribute workloads efficiently among ESs. An unbalanced workload can

lead to some servers being overwhelmed while others are underutilized. Third, it should keep providing services irrespective of occasional ES failures. Imagine a big city with hundreds of Base Stations (BSs) serving mobile users. The placement of ESs in this complex network is a puzzle. If we get it right, mobile applications will run smoothly, even if one ES goes offline temporarily. If we get it wrong, we might see delays, work overload, and even service interruptions.

Our thesis is all about finding the best way to solve this puzzle, placing ESs strategically. We are using sophisticated algorithms and real-world data to make sure mobile users get fast, reliable access to services, and the system stays resilient, even in challenging situations. Making MEC resilient and dependable for everyone is the focus of our work. Building on these insights, the next section outlines our specific research objectives, focusing on designing an optimized placement strategy for ESs within MEC networks.

1.2 Research Objectives

In the context of this research, our foremost objective is to meticulously design an optimized placement strategy for ESs within MEC networks. This strategy is driven by the core principles of load balance, access delay reduction and resilience. The primary aim is to achieve an equitable distribution of computational workloads among ESs, preventing undue strain on any particular server. Furthermore, we are dedicated to minimizing the access delay experienced by mobile users when interacting with ESs. This reduction in latency is pivotal in enhancing the overall Quality of Service (QoS) for end-users. Moreover, our research goes beyond the conventional paradigms. We are committed to ensuring that the benefits of load balance and latency reduction persist even in the face of ES failures or disruptions. Resiliency is at the heart of our placement strategy, enabling MEC networks to adapt swiftly and effectively to unforeseen circumstances. Whether it is an ES malfunction, network congestion, or abrupt changes in demand, our approach is designed to maintain acceptable performance levels and ensure uninterrupted service delivery.

In summary, our research objectives are centered on the development of a robust and forward-looking Edge Server Placement (ESP) strategy. This strategy focuses on two key pillars: workload balancing to optimize resource utilization and minimizing access delay for mobile users. Furthermore, the strategy incorporates resiliency measures to safeguard against disruptions and failures, ultimately guaranteeing an exceptional QoS in MEC net-

works, regardless of the challenges encountered. Having outlined our research objectives, we now turn to the specific contributions of this thesis, which address the challenges identified and aim to achieve the outlined goals.

1.3 Contributions

In this thesis, our contributions are summarized as follows.

1. *Single Objective Constraint Optimization*: We propose an innovative approach to address the ESP problem as a single-objective constraint optimization problem.

2. *Efficient Placement Algorithm*: We develop an efficient algorithm that determines the strategic placement of ESs. This algorithm considers factors such as workload balancing among ESs, minimizing access delays, and resilience. Our proposed mathematical formulation simultaneously takes into account various components, including BS traffic loads, ES capacities, and latencies between BS-ES and ES-ES. Additionally, we ensure uninterrupted service to mobile users in the event of an ES failure by assigning multiple ESs to a single BS. This redundancy enhances fault tolerance and improves the reliability of the MEC network. Additionally, we recognize that measuring QoS and maintaining Service Level Agreements (SLAs) are crucial factors for network service providers. Our proposed scheme assists in achieving these performance parameters by establishing latency bounds for BS-ES and ES-ES communication.

3. *Experimental Validation*: To validate our approach, we conduct experiments using a dataset comprising approximately 3000 BSs operated by Shanghai Telecom. Our results demonstrate the superiority of our methodology compared to other approaches.

These contributions collectively enhance our understanding of ESP in MEC networks and pave the way for more resilient, efficient, and high-performing network architectures.

1.4 Thesis Outline

The outline of this thesis is as follows.

Chapter 1 introduces the concept of MEC, explains the research objectives, and discusses the contributions of this thesis. It concludes with an overview of the thesis structure.

Chapter 2 provides a comprehensive background and review of related works relevant to our research topic.

Chapter 3 focuses on the placement of ESs while considering resiliency. It introduces the resilient ESP problem in MEC, providing a detailed system model and problem formulation. The chapter also presents our proposed heuristic algorithm for solving the problem, along with an analysis of its complexity.

Chapter 4 presents the results of extensive numerical evaluations, comparing the performance of our proposed algorithm wtih existing ESP methods. This chapter utilizes real-world datasets for evaluation and provides detailed discussions on the experimental setup and results.

Chapter 5 concludes the thesis by summarizing the research findings, drawing key conclusions, and outlining potential avenues for future work.

Chapter 2

Background and Related Works

This chapter provides a foundational understanding of the concepts and research relevant to our study. Section 2.1 explores the background of MEC, detailing its purpose, growth, impact on user experience, and industry relevance, and contrasting it with traditional cloud computing. Section 2.2 reviews existing literature and research on key areas related to our work, including the optimization of ESP, its application in various domains, resilience in different computer networks and edge computing, and energy efficiency. Section 2.3 discusses key concepts and methodologies that support our research, such as Facility Location and Clustering Algorithms, which are crucial for developing our proposed solution. This comprehensive review and conceptual exploration provide the groundwork for a closer examination of the challenges and advancements in MEC, informing the context of our research.

2.1 Background

MEC, a relatively recent technological development, is a natural progression in the evolution of mobile BSs and the convergence of IT and telecommunications networking [3]. In the dynamic landscape of MEC, the strategic placement of ESs plays a pivotal role in shaping the performance, reliability, and resilience of MEC networks. MEC has emerged as a transformative paradigm, addressing the escalating demand for low-latency and high-performance computing within wireless networks. By relocating computation and storage capabilities closer to the network's periphery, MEC not only enables efficient data processing but also facilitates the implementation of once-impossible latency-sensitive applications, transcending the limitations of conventional cloud computing.

2.1.1 Purpose of MEC

MEC, which began gaining traction in the early 2010s, focuses on processing data from mobile devices and applications closer to end-users [4]. It encompasses diverse contexts such as video cameras, mobile or remote medicine, IoT in all its iterations, gaming (including AR/Virtual Reality (VR)), connected vehicles, and more [5]. The driving forces behind the creation of MEC include the increasing demand for low-latency applications, the need for real-time data processing and analysis, and the limitations of cloud computing. MEC brings computational and storage resources to the edge of the mobile network, allowing demanding applications to be executed within the MEC infrastructure rather than on User Equipment (UE) or centralized cloud servers while ensuring minimal latency thresholds [6]. This setup reduces latency, optimizes power consumption for local tasks, and enhances cybersecurity by enabling real-time threat detection and secure data handling [3, 7]. In summary, MEC is not just a technological evolution; it is a response to the increasing demand for low-latency applications, real-time data processing, and the limitations of traditional cloud computing. This transformative technology is making waves across industries, bringing low-latency and high-bandwidth services, real-time data processing, and enhanced cybersecurity.

2.1.2 Growth of MEC

Compelling statistics and data showcase the growth of MEC, the rising demand for low-latency applications, and the challenges faced by MEC networks. MEC has emerged as a major computing paradigm with the evolution of 5G and Internet of Things (IoT) technologies [8]. It addresses the resource limitations of mobile devices and the bandwidth bottleneck of the core networks in mobile cloud computing [9]. Challenges include mobility issues, fluctuations in wireless channel quality, and dynamic changes in device energy levels [8, 9]. MEC's capability to push computation and storage to the network's edge allows for the execution of resource-intensive applications close to the end-users, ensuring minimal latency and meeting strict delay requirements [10]. The escalating demand for low-latency applications is met by MEC, significantly reducing latency and optimizing the power consumed by tasks executed locally [11]. These capabilities highlight the significance of MEC in the current technological landscape. While it promises a transformative computing paradigm, the challenges, particularly in mobility and network fluctuations, need careful consideration.

2.1.3 User Experience

User experience is a critical facet of MEC networks. Latency and reliability directly impact users' experiences in applications like AR, autonomous vehicles, or real-time video streaming. MEC's approach of bringing computation and storage resources to the network's edge ensures the execution of demanding applications in the UE, meeting stringent delay requirements [8]. MEC not only significantly reduces latency but also optimizes power consumption for locally executed tasks [12]. In applications like AR, low latency is essential to ensure that virtual objects align correctly with real-world objects. In autonomous vehicles, low latency is critical for real-time environmental responsiveness, and in real-time video streaming, it ensures smooth, uninterrupted viewing [13]. Reliability is paramount in MEC networks due to their dynamic nature. This dynamic nature leads to fluctuations in wireless channel quality, variable computation resource availability in ESs, and unpredictable changes in device energy levels [8]. Ensuring consistent performance is essential to prevent service disruptions and uphold the user experience. In summary, user experience is a central concern in MEC networks. MEC's ability to reduce latency and optimize power consumption contributes significantly to user satisfaction. The dynamic nature of MEC networks necessitates a focus on reliability to ensure consistent, high-quality performance.

2.1.4 Industry Relevance

MEC networks are not confined to the realm of theory; they hold real-world relevance across diverse industries. Several examples illustrate how MEC technologies are transforming operations. MEC optimizes production processes by providing real-time monitoring and control of industrial processes, as well as predictive maintenance and quality control. Connected IoT devices exist throughout the process, generating large amounts of data that can be processed at the edge—meaning the data is analyzed and acted upon right where it is generated, enhancing efficiency and responsiveness [14]. MEC enhances agricultural operations through real-time monitoring of crops and livestock, enabling precision agriculture—an approach that leverages advanced technology for optimized farming practices. This approach includes smart irrigation and utilizing real-time weather data to make informed decisions about irrigation, fertilization, pest control, and overall crop management [14]. MEC plays a crucial role in enabling 5G-based health monitoring systems for the Internet of Medical Things (IoMT) [15]. It facilitates decentralized game-theoretic approaches for resource allocation and task offloading in IoMT networks, ensuring efficient and secure healthcare services. MEC's low latency and high bandwidth capabilities are

vital for real-time monitoring, data processing, and decision-making in healthcare applications. MEC enhances the driving experience by providing real-time traffic information, navigation, and a range of entertainment services. This entertainment services include features such as in-car streaming, personalized music recommendations, and interactive infotainment options, all delivered seamlessly through MEC technology. It also supports autonomous driving through low-latency, high-bandwidth services [14]. MEC supports immersive technologies like VR and AR by providing the necessary computing resources at the edge. An edge-computing based architecture has been proposed for mobile AR, leveraging MEC's low latency and high bandwidth to deliver seamless AR experiences on mobile devices [16]. MEC enables offloading compute-intensive tasks like 3D rendering and object recognition to ESs, enhancing the performance and battery life of mobile AR/VR applications. MEC is an emerging technology in the telecommunications industry, enabling the deployment of new services such as AR, VR, and autonomous vehicles. It enhances cybersecurity through real-time threat detection and response [3, 6]. MEC brings cloud-computing capabilities to the edge of the mobile network, enabling real-time access to radio network information and providing secure data storage and transmission. These capabilities are characterized by ultra-low latency and high bandwidth, leveraging the unique advantages of cloud computing at the network edge [3, 9].

These examples highlight how MEC networks are driving transformation across various sectors, offering low-latency, high-bandwidth services, real-time data processing, and enhanced cybersecurity. The real-world relevance of MEC is expected to grow in the evolving technological landscape. In the following section, we explore the unique features of MEC by comparing it to traditional cloud computing, shedding light on the distinctive advantages that MEC brings to the forefront.

2.1.5 Comparison to Traditional Cloud Computing

A crucial consideration in understanding MEC is contrasting it with traditional cloud computing. While traditional cloud computing relies on centralized data centers for data processing, MEC brings computational and storage capabilities closer to the network's edge, allowing demanding applications to operate near the end user, thus meeting stringent latency requirements [8]. The limitations of traditional cloud computing for latency-sensitive applications emphasize the necessity of MEC. Traditional cloud computing, with its centralized data processing, is unsuitable for mission-critical mobile applications (e.g., AR/VR/

Mixed Reality (MR), autonomous driving) with strict low end-to-end latency requirements [8]. For instance, in scenarios like autonomous driving, the time taken by traditional cloud computing to relay information to centralized data centers, sometimes up to 2 seconds, introduces delays in the decision-making process [17]. This latency can result in critical consequences, such as accidents or operational inefficiencies, leading to significant losses for organizations, making MEC a preferred choice. MEC not only significantly reduces latency but also optimizes the power consumed by locally executed tasks [10]. Additionally, MEC enhances cybersecurity through real-time threat detection and response, along with secure data storage and transmission [9]. In summary, MEC stands out as a more suitable approach for latency-sensitive applications compared to traditional cloud computing. By bringing computation and storage resources to the edge of the mobile network, MEC not only meets strict latency requirements but also offers advantages in power optimization and cybersecurity. However, MEC environments are more dynamic and prone to faults/fluctuations than traditional cloud environments [8].

At the heart of this dynamic ecosystem are ESs, strategically placed to orchestrate the seamless delivery of services. However, MEC networks, given their distributed nature, are susceptible to disruptions like ES outages, network congestion, and fluctuating traffic patterns. These challenges can potentially disrupt the functionality and availability of MEC services. For instance, in scenarios with sudden spikes in user demand or during natural disasters, disruptions in ES placement may lead to service outages. Given this vulnerability, ensuring the resilience of ES placement becomes paramount—a facet often underexplored in existing research.

Resilience, in the context of MEC, involves fortifying the network against adversities, ensuring it can withstand unforeseen contingencies and continue functioning with minimal interruption. This research aims to bridge this knowledge gap by developing resilient strategies for ES placement within MEC networks. The objective of this research is to develop an efficient and resilient ES placement approach. This method not only mitigates latency and balances workloads but also enhances the overall resilience of MEC networks. The technique should be dynamic and adaptable, capable of responding to the unpredictable nature of MEC environments. With this objective in mind, the following section explores existing research works that have contributed to advancements in ES placement within MEC networks.

2.2 Existing Research Works

In exploring the current landscape, it is crucial to examine related works that have significantly advanced our understanding of ES placement in MEC networks. Recent years have seen substantial attention on optimizing ES placement within MEC, with a focus on enhancing key performance metrics such as latency, reliability, and cost-effectiveness. Numerous studies have explored this complex challenge. This section provides a comprehensive review of existing research, highlighting key contributions and approaches, and identifying significant gaps in the literature. By contextualizing our research within this landscape, we aim to position our work effectively and identify opportunities for novel contributions in the field.

2.2.1 Optimization Strategies for ESP

A significant thread of research has focused on refining ES deployment strategies for improved system reliability and performance. Optimization algorithms play a crucial role in determining the optimal deployment locations for ESs in MEC networks. Building on these optimization strategies, Wang et al. [18] introduce a fault-tolerant server deployment model and an Improved Grey Wolf-Genetic Algorithm for optimal ES deployment in the industrial internet, showcasing improvements in system reliability and performance. This approach aligns with the work of Jian et al. [19], who leverage the k-means algorithm for optimized cloud ES placement, emphasizing load balancing and response time optimization. Their approach clusters system sources and application loads, selecting the most suitable server for each application. Additionally, addressing the dynamic nature of ESP, Jiang et al. [20] tackle the dynamic nature of ESP with algorithms based on Deep Reinforcement Learning (DRL). Adapting to changing network states and placement costs, their dynamic algorithms outperform counterparts by significant margins. Complementing this achievement, Luo et al. [21] introduce DQN-ESPA, a novel ESP algorithm based on DRL. This algorithm models the problem as a Markov decision process, outperforming existing methods in terms of optimal placements without relying on previous experience. Expanding on these studies, Li et al. [22] propose a clustering-based approach for optimizing the deployment of Mobile ESs (MES) in MEC scenarios, improving network performance and reducing completion time, power consumption, and overhead. Hu et al. explore ESP in the context of genetic algorithms for MEC, discussing latency reduction and contextual awareness improvement [23]. In addressing load balancing and access latency optimization in MEC,

Chen et al. tackle load balancing and access latency optimization, proposing an immune optimization algorithm to improve resource utilization and meet user requirements [24]. Cao et al. [25] explore the placement of heterogeneous ESs in mobile edge-cloud computing systems to minimize response time. Their approach reduces system response time by 47.37% and improves BS response time fairness by 71.60%. Gong et al. [26] introduce an ILP-based algorithm to minimize access delay by optimizing the placement of ESs in MEC networks. Moreover, Qu et al. [27] propose a robust submodular maximization approach for server placement in edge computing, considering uncertain ES failures and aiming to maximize the expected overall workload. Guo et al. [1] propose a user allocation-aware edge cloud placement approach in MEC to balance workload and minimize communication delay. They formulate it as a multiobjective optimization problem and propose an approximate solution using K-means and mixed-integer quadratic programming. In addition to the aforementioned studies, Cui et al. [28] introduce a joint user coverage and network robustness oriented ESP strategy in edge cloud computing, considering the trade-off between user coverage and network resilience. Yin et al. [29] propose Tentacle, a decision support framework that optimizes ESP for online service providers, considering performance, cost, and pragmatic concerns. While these studies employ diverse optimization strategies, the common thread lies in their collective goal of improving the efficiency and effectiveness of ESP in MEC networks.

Surveys and overviews provide a comprehensive understanding of the existing solutions and research gaps in ESP. Lahderanta et al. present a survey on existing solutions, identifying research gaps and proposing the PACK algorithm, which considers various parameters for optimal server placement and workload allocation [30]. Bahrami et al. [31] provide a comprehensive overview of the ESP Problem (ESPP) in Multi-Access Edge Computing (MEC) environments. Their work serves as a valuable resource, offering insights into various research approaches, challenges, and optimization strategies related to ESP. To further explore the application of ESP techniques in specific contexts, the following subsection explores ESP techniques tailored to address unique requirements and challenges within specific domains.

2.2.2 ESP in Specific Domains

ESP techniques are tailored to specific domains to address unique requirements and challenges, reflecting a diverse array of research efforts aimed at optimizing deployment strate-

gies. Studies across various domains share common goals of improving system reliability, performance, and efficiency through strategic ESP. In mobile environments, researchers have explored location-aware ESP strategies. Shao et al. [32] consider factors such as user distribution density, deployment cost, and network access delay in wireless metropolitan area networks, reflecting a practical approach to ES deployment. Similarly, for smart cities, Wang et al. [33] and Zhao et al. [34] emphasize workload balancing and minimizing access delay, showcasing the growing importance of MEC in urban computing infrastructure. Hybrid strategies for optimal deployment have gained attention, integrating fault-tolerant server deployment models with optimization algorithms. Wang et al. [35] incorporate fault tolerance into their ESP model for intelligent manufacturing, demonstrating significant improvements in load balancing and time savings. This hybrid approach indicates the potential of integrating fault tolerance mechanisms into deployment models to enhance system reliability. In domains such as the Industrial Internet of Things (IIoT) and cellular networks, researchers explore heuristic and dynamic server placement approaches. Kasi et al. [36] propose genetic algorithms and local search techniques for optimal ESP solutions in IIoT, while Shen et al. [37] contribute a dynamic placement approach for edge computing in the Internet of Vehicles (IoV), aiming to enhance quality of service and minimize reconstruction costs. Furthermore, comprehensive optimization models have been developed for ES deployment in dynamic vehicular environments. Cao et al. [38] present a holistic six-objective optimization model considering factors such as transmission delay, workload balancing, energy consumption, and network reliability. This comprehensive approach reflects the intricacies of deploying ESs in dynamic vehicular environments, addressing the multifaceted challenges of ESP in IoV scenarios. In the domain of ESP, several studies aim to optimize resource allocation and improve system performance in diverse edge computing environments. Farhadi et al. [39] and Li et al. [40] both address the optimization of service placement, albeit in different contexts. While Farhadi et al. focus on data-intensive applications and propose a two-time-scale framework for optimization, Li et al. concentrate on ultra-dense networking environments, aiming to minimize service provider costs and guarantee completion time. Hadvzic et al. [41] and Zhu et al. [42] both tackle challenges in specific edge computing architectures. Hadvzic et al. delve into the telecom-centric Multi-Access Edge Computing (MEC) architecture, challenging assumptions regarding negligible latency at the edge and proposing strategies for packet gateway distribution. Meanwhile, Zhu et al. address the maintenance of high availability in a less robust MEC cloud environment, specifically in 5G networks, by considering resource cost and application availability. Ahat et al. [43] contribute to the optimization of ES deploy-

ment across multiple tiers in cloud computing networks. By proposing a Mixed-Integer Linear Programming (MILP) model and heuristic algorithms, they aim to enhance resource allocation efficiency and contribute to ongoing efforts in optimizing edge resource deployment. In summary, while each study addresses different aspects of ESP and optimization, they collectively highlight the domain-specific challenges and offer tailored deployment strategies. By synthesizing various approaches, researchers advance the field of ESP in MEC networks, enabling more efficient deployments. These strategies also set the stage for addressing resilience in different computer networks, a critical aspect explored in the next section.

2.2.3 Resilience in Different Computer Networks

The synthesis of studies on resilience across various computer networks reveals a multifaceted landscape of research efforts aimed at enhancing the reliability and robustness of distributed computing environments. Surveys and knowledge systematization have played crucial roles in helping us understand the landscape of resilience across diverse computing environments. Amiri et al. [44] conduct a systematic literature review, providing a comprehensive overview of resilient and dependable management approaches in distributed environments. This literature review includes cloud, edge, fog, IoT, Internet of Drones (IoD), and IoV. The review proposes future research directions, analyzing challenges and strategies. Similarly, Berger et al. [45] systematically organize knowledge regarding efforts to enhance the resilience of IoT systems. This knowledge includes taxonomy, classification, and state-of-the-art resilience mechanisms. Castillo et al. [46] offer an insightful overview of resilience in computer networks. This overview encompasses the analysis of network topologies and metrics used to measure resilience. They propose a model based on resilience metrics, discussing the behavior of resilience in different network topologies, while Welsh et al. [47] provide an overview of resilience techniques in cloud environments, including emerging paradigms like fog and edge computing. In industrial contexts, Fowler et al. [48] emphasize the importance of resilience in smart factories enabled by digital technologies, while Sahoo et al. [49] provide a comprehensive survey of replica server placement algorithms in traditional and emerging paradigms for Content Delivery Networks (CDNs). In addition, Jung et al. [50] propose ShadowBox, a novel redundancy-aware VM scheduler that optimizes the placement and activation of standby VMs in cloud environments. This improves resource utilization while ensuring applications' resource entitlements. In telecommunications, Esmat et al. [51] discuss network slicing solutions for

satellite-terrestrial edge computing networks (STECNs) and emphasize the importance of resilient network slicing. Thiruvasagam et al. [52] propose a novel approach using multi-connectivity in 5G networks to address failures in network slices, MEC cloud servers, and communication links. This approach ensures resilience and high availability of services. Manias et al. [53] provide a robust optimization model for post-fault VNF placement in 5G networks. This model addresses reliability and resilience while preserving QoS. Hyodo et al. [54] discuss a resilient VNF placement model that minimizes the cost of using computation resources while guaranteeing recovery against single facility node failures within specified recovery time objectives (RTOs). In radio access networks, Xing et al. [55] introduce Atlas, a system providing resilience for the Distributed Unit (DU) in virtualized radio access networks (vRANs) by repurposing existing cellular mechanisms. Nasralla et al. [56] investigate the impact of power outages on telecommunication network performance, proposing a Mixed Integer Linear Programming (MILP) model for evaluating network performance during a blackout. Kaleem et al. [57] propose a disaster-resilient three-layered architecture for Public Safety Long-Term Evolution (PS-LTE) communication, integrating SDN, Unmanned Aerial Vehicles (UAV), cloudlet, and radio access layers to meet the delay requirements of emergency services. Addressing security concerns, Azab et al. [58] present a spatiotemporal runtime diversification approach to enhance the security and resilience of Software-Defined Networking (SDN) controllers in Smart Grid networks. They address the Controller Placement Problem (CPP). Aibin et al. [59] discuss security challenges in Software-Defined Networks (SDNs), CDNs, and Information-Centric Networks (ICNs). They propose solutions to address these challenges and examine the potential for updating communication networks to reduce vulnerability to attacks. Lastly, Tanha et al. [2] discuss a resilient controller placement problem for SD-WANs. This considers switch-controller latency, controller capacity, and traffic load. They provide high-quality solutions using clique-based algorithms. While existing studies cover resilience in various contexts, they lack a focused exploration of resilient ESP within MEC networks. Our work fills this gap by highlighting resilience in ESP, a largely overlooked area in current literature. Focused research on resilience in edge computing systems, particularly MEC networks, is still needed.

2.2.4 Resilience in Edge Computing Systems

In the realm of resilience in edge computing systems, researchers have explored various approaches to enhance the robustness and reliability of edge computing environments against

attacks, faults, and uncertainties. Cheng et al. [60] propose a two-stage robust model focused on resilient service placement and workload allocation. Their proactive measures aim to mitigate uncertainties, ensuring uninterrupted operations and preserving service quality. Talpur et al. [61, 62] address security concerns in the IoV by proposing attack-resilient mapping of vehicles to edge nodes and a service placement framework. Their Deep Reinforcement Learning (DRL) framework optimizes Vehicle-to-Edge (V2E) mapping to maintain service availability and minimize disruption in the face of attacks. Siew et al. [63, 64] contribute algorithms integrating importance sampling into actor-critic reinforcement learning and introduce the Failure-Adaptive Reinforcement Learning (FIRE) framework, designed for edge computing migrations and considering rare events such as server failures. Moura et al. [65] introduce RECS, a Resilient Programmable System for Edge Cloud Systems, enhancing operational resilience against faults, congestion, and cyber-attacks. In IoT applications in MEC, Pietrantuono et al. [66] propose a testing-based methodology to assess the impact of resource exhaustion attacks and identify resilience-related indicators. Masoumi et al. [67] propose a dynamic Virtual Network Function (VNF) placement algorithm in MEC environments, considering different protection schemes to ensure resilience against failures. Isa et al. [68] propose a resilient fog computing infrastructure for healthcare monitoring, incorporating server and network protection and optimizing server placement to minimize energy consumption. Discussing the state of the art in Fog Computing Systems (FCS), Moura et al. [69] emphasize resilience as a critical aspect, addressing current research issues and forecasting future trends. Cui et al. [70] tackle the Robustness-oriented k Edge Server Placement (RkESP) problem, providing optimal and approximate approaches to address ES failures in MEC scenarios. Ford et al. [71] propose algorithms for provisioning distributed Mobile Edge Clouds (MECs), aiming for lower latencies, increased resilience, and cost savings in mobile networks. Shirazi et al. [72] discuss the emergence of MEC and Fog as extensions of the cloud, emphasizing their impact on communication and networking service models and emphasizing the crucial importance of security and resilience. Colman et al. [73] provide a comprehensive overview of resiliency techniques in cloud computing, addressing failures at the server, network, and application levels, offering valuable insights into building resilient cloud architectures. Overall, these studies advance resilience in edge computing systems by proposing various methodologies and frameworks to address specific challenges and uncertainties. However, while existing research has enhanced our understanding of fault tolerance and robustness, many approaches lack a comprehensive strategy for maintaining continuous service during ES failures, especially under various constraints. Our work addresses this gap by focusing on ESP that

ensures resilience, maintains uninterrupted services, and optimizes network performance even when ES malfunctions occur. Moving forward, we explore the critical aspects of energy efficiency and cost optimization in ESP, essential for the sustainability and effectiveness of edge computing environments.

2.2.5 Energy Efficiency and Cost Optimization in ESP

The studies in this subsection focus on energy efficiency and cost optimization in ES placement, highlighting their key role in improving performance and economic viability in edge computing. As indicated by Dayarathna et al. [74], the primary source of energy wastage stems from servers remaining idle. Thus, enhancing equipment utilization is crucial to minimizing idle server instances. Li et al. [75] and Zhang et al. [76] both address the profit maximization aspect of ESP, considering factors such as access delay, energy consumption, and deployment costs. Li et al. [77] introduce energy-aware ESP algorithms to reduce energy consumption, while Fan et al. [78] propose the CAPABLE strategy to optimize the tradeoff between deployment cost and end-to-end delay. Yang et al. [79] and Wang et al. [80] focus on maximizing coverage area and cost-effectiveness under budget constraints, respectively. Su et al. [81] contribute an online algorithm to minimize operational costs and address time-correlated service placement costs. Ren et al. [82] and Zeng et al. [83] both propose low-cost ESP strategies in wireless metropolitan area networks, aiming to minimize the number of servers while meeting quality of service requirements. Overall, these studies collectively advance the understanding of energy efficiency and cost optimization in ESP, providing diverse methodologies and algorithms to address the critical intersection of economic viability and performance enhancement in ESP. As we explore strategies to minimize energy consumption and maximize cost-effectiveness in ESP, it becomes evident that these efforts are integral to achieving sustainable and efficient edge computing infrastructures. One of our objectives is to minimize workload variance among ESs, ensuring efficient utilization across the network. By preventing underutilization in some servers while others become overloaded, our approach enhances overall server utilization and energy efficiency within the MEC network. Having reviewed the relevant existing research, we now shift our focus to the key concepts and methods that form the foundation of our proposed solution.

2.3 Key Concepts and Methods

To effectively address the challenges of ESP in MEC networks, we base our solution on two foundational concepts: Facility Location and Clustering Algorithms. These concepts are critical for determining strategic ES locations and efficiently assigning BSs to these ESs. In the following subsections, we discuss these key concepts, highlighting their relevance and application to our proposed solution.

2.3.1 Facility Location and Its Relevance to Our Problem

The Facility Location Problem (FLP) is a well-known optimization challenge with a broad range of applications [84, 85], including its relevance to ESP in MEC networks. In both FLP and ESP, the objective is to determine strategic locations for facilities (ESs) to serve demand points (mobile devices) effectively. Key parallels include workload distribution, where FLP minimizes transportation costs, and ESP ensures balanced workloads for low latency. Both problems also involve location selection to minimize distance and resource allocation to meet demand efficiently. Moreover, resiliency is considered in both contexts, with FLP establishing redundant facilities and ESP maintaining connectivity between BSs and multiple ESs to ensure service continuity. These parallels enable the use of optimization techniques from FLP to develop efficient solutions for MEC networks.

2.3.2 Clustering Algorithms and Their Relevance to Our Problem

Clustering algorithms are fundamental for grouping data points into clusters based on similarities and have applications in various domains. Among them, the K-Medoids algorithm stands out as particularly relevant to our optimization of ES placement (ESP) in MEC networks. Clustering algorithms reveal the underlying structure in datasets by grouping similar data points together, with applications in machine learning, data mining, and statistics. The K-Medoids algorithm is a notable technique that extends the concept of K-Means clustering [86]. While K-Means uses the mean of data points to represent cluster centers, K-Medoids employs the medoid—the most centrally located data point within a cluster [87]. This approach is more robust to outliers and noise in the data.

The K-Medoids algorithm proceeds through the following steps: initialization by randomly selecting K data points as initial medoids; assignment of each data point to its nearest medoid, creating K clusters; update of medoids to ensure they are as close as possible

to other data points within the cluster; iterative refinement of medoids until convergence; and final determination of the medoids and their respective clusters as the algorithm's output [88]. The relevance of K-Medoids algorithm to our ESP problem lies in its efficiency in identifying strategic ES locations, considering medoids as potential ES locations. This strategic positioning minimizes access delay for mobile devices, a critical factor for delivering low-latency services in MEC networks.

2.4 Conclusion

In summary, this chapter lays the foundation for our research work in this thesis. This foundation emphasizes the critical role of resilience in ES placement, aiming to fill the research gap in this domain. Furthermore, this chapter provides a survey on the state-of-the-art methods and algorithms presented in the literature, focusing on improving the resilience, reliability, and overall performance of MEC networks. This survey acknowledges the unpredictable nature of MEC environments and aims to improve the robustness of ES placement for an enhanced user experience.

Despite extensive studies on various aspects of ESP and optimization, significant research gaps remain. Resilience in ESP is often underemphasized, risking network disruptions and deteriorated service quality. Additionally, many studies concentrate on specific domains like manufacturing or smart cities, lacking a holistic approach adaptable to diverse scenarios. The absence of a unified framework that integrates both communication and computation models while strategically deploying ESs in densely populated areas further highlights the need for comprehensive research in this field. To address these gaps, our research introduces a novel approach that combines the K-medoids clustering algorithm for ESP with a heuristic algorithm for BS allocation. This method offers advantages over existing approaches, even in scenarios involving ES failures. Optimizing ES utilization is also crucial to minimize energy wastage from idle servers. By addressing varying computing demands across urban areas, our strategy not only optimizes energy consumption but also allocates computing resources efficiently. Consequently, adopting strategies that minimize access delay and workload variance, while aligning with resilience objectives in MEC networks, improves network performance and energy efficiency. The effectiveness of our approach is demonstrated through experiments with real-world data, showing reduced delays and uninterrupted service delivery. As we transition into Chapter 3, we will explore the problem formulation and our proposed solution to address the critical challenges

identified in the preceding chapters.

Chapter 3

Resilience in Edge Server Placement in Mobile Edge Computing Networks

In the previous chapter, we explored various aspects of ESP. Building on that foundation, we now turn our focus to examining ESP in MEC networks, with particular emphasis on resilience. Section 3.1 introduces our system model, outlining the preliminaries, assumptions, and the communication and computation models. We also discuss resource allocation strategies aimed at enhancing resilience. Section 3.2 formulates the problem, providing a basis for our proposed solutions. The concepts from facility location and clustering algorithms, discussed in the previous chapter, form the foundation for the solution described in Section 3.3. Finally, we detail our algorithm and analyze its complexity, demonstrating its effectiveness in optimizing ESP in MEC networks.

3.1 System Model

3.1.1 Preliminaries

Our investigation into the resilience of ESP, and the ESP problem itself, can be viewed as variants of the well-known FLP. In the context of a MEC network, ESs function as facilities, while BSs serve as clients. We assume a finite number of BSs with varying service demands and a finite number of ESs and potential ES locations. The main challenges in this scenario are determining (1) the most suitable locations for placing ESs and (2) the assignment of BSs to these ESs, a problem known for its computational complexity [33].

Figure 3.1 illustrates an example of ESP in an MEC network. Mobile users send service re-

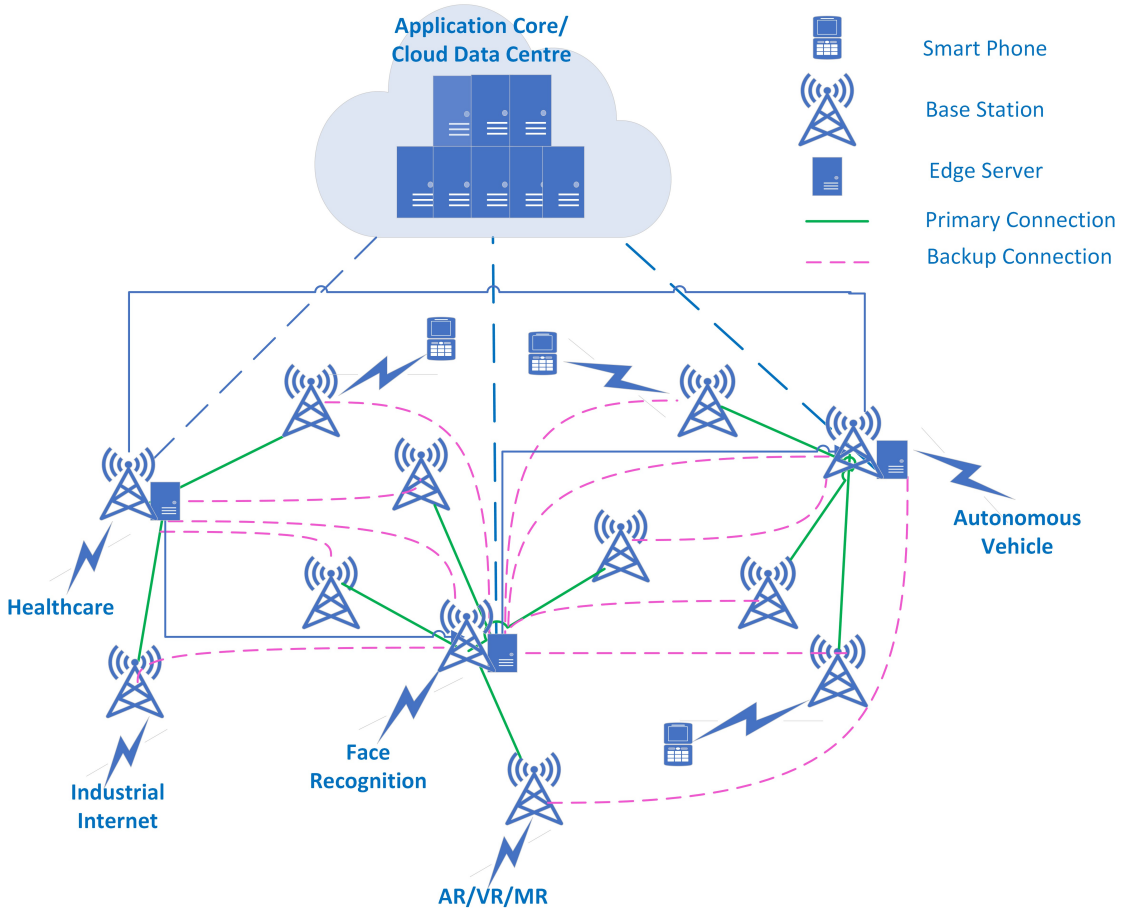


Figure 3.1: An example of ESP in MEC.

quests to a BS with fixed radio coverage. Each BS forwards these requests to its associated ESs. The total number of requests directed to an ES represents its workload. We assume K ESs are placed at K different locations, each with equal computing resources. For request processing, each BS has one primary ES and a secondary ES for backup. We need to optimize the placement of these ESs among BSs to minimize access delay. Access delay is proportional to the distance between the BS and ES. Additionally, it is crucial to define the dominant area for each ES to ensure balanced workloads. Workloads are balanced by distributing them equitably across the ESs. Each BS should also be associated to a backup ES to maintain service continuity in case of primary ES failure.

3.1.2 Our Assumptions

We outline several key assumptions that guide our research:

- **Node failures (single or multiple ES):** Our focus is solely on ES node failures within the MEC network. We exclude ES site failures, which result from the complete destruction of the BS site(s) where the ES is located. These site failures, although significant, are less frequent and are not within the scope of our research. To enhance MEC network resilience, we implement a primary/backup model where each BS has a primary ES responsible for serving its requests, along with one or more backup ESs. While each ES can serve as both primary and backup for different BSs, the same ES cannot serve both roles for a single BS. This design choice highlights the principles of diversity and separation, which are vital for ensuring BS-level resiliency. By dedicating specific primary and backup ESs to each BS, we aim to enhance the resilience of the network, ensuring uninterrupted service even in the event of an ES node failure. This strategy safeguards service continuity at the BS level, aligning with our objective of achieving BS-level resiliency in the MEC network.

- **Objectives of the optimization problem:** Our objective in addressing the optimization problem is to achieve a balance in workload distribution among ESs, minimize access delay between BSs and ESs, and enhance network resilience. To accomplish these goals, we focus on optimizing workload distribution among ESs and introducing QoS parameters. These parameters allow network designers to customize various aspects, such as BS-ES and ES-ES latency, to meet specific requirements. By doing so, they can optimize cost, reliability, and performance across the network.

- **Delay bounds (BS-ES, ES-ES):** We set upper limits on both BS-ES and ES-ES latencies to ensure that the network performance aligns with acceptable standards. This action is pivotal because the strategic placement of ESs significantly impacts both the access delays experienced by clients and the efficient utilization of ES resources. For instance, ESs' close proximity to one another reduces ES-ES latency but might lead to higher BS-ES latency. Conversely, geographically dispersed ESs lower BS-ES latency but increase ES-ES latency. To address this challenge, we introduce latency bounds, which are crucial for latency-sensitive applications. By setting upper limits on both BS-ES and ES-ES latencies, we guarantee network performance. Prioritizing BS-ES latency to be less than or equal to ES-ES latency ensures that communication between BSs and their assigned ESs is efficient and timely, which is vital for delivering prompt services to mobile users. This requirement helps maintain a balance that enhances overall network performance and user experience.

- **Redundancy and fault tolerance through primary and backup connections:** We establish both primary and backup connections for each BS, ensuring that the same ES is not serving as both the primary and backup for a particular BS. This redundancy enhances fault tolerance by providing uninterrupted service.

Having established the assumptions, we now examine the communication and computation models within the MEC network.

3.1.3 Communication Model

In the MEC network, the communication model plays a pivotal role in shaping interactions between mobile users, BSs, and ESs. This model delineates how data and requests flow within the network, providing crucial insights into optimizing ESP.

3.1.3.1 User-to-BS Communication

Mobile users access services by sending requests to the nearest BS. Although our primary focus is on the communication links between BSs and ESs, as well as among ESs, it is important to understand how user-to-BS interactions impact the overall MEC network. Key aspects include *Request Routing* based on signal strength and coverage area, *Propagation Delay*, and *Workload Distribution* among BSs, which is essential for load balancing and minimizing access delay.

3.1.3.2 BS-to-ES Communication

Each BS communicates with one or more ESs to handle user requests, including data transmission, processing, and result delivery. Key considerations are ensuring each BS connects to at least two distinct ESs for redundancy, minimizing latency to reduce access delay, and maintaining network resilience through backup mechanisms to mitigate ES failures.

3.1.3.3 ES-to-ES Communication

ESs may communicate with each other for load balancing and backup purposes. Load balancing algorithms enable ESs to evenly distribute workloads, preventing bottlenecks. In the event of an ES failure, backup ESs assume partial responsibilities to maintain network service levels and minimize disruptions.

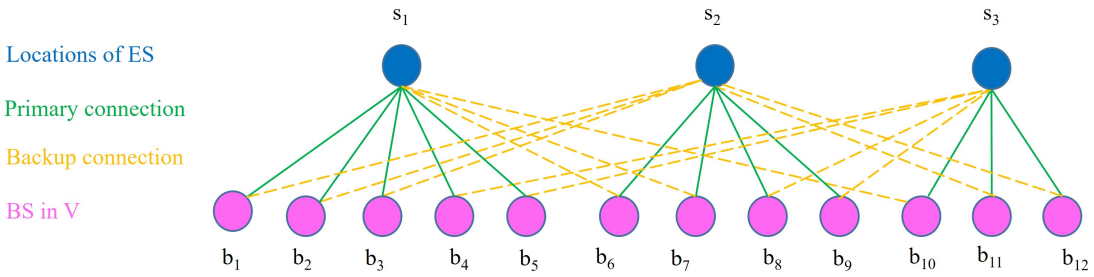


Figure 3.2: A sample scenario showing the assignment of BSs to ESs.

We emphasize the importance of accurately modeling communication aspects for an effective ESP strategy. The choice of ES locations significantly impacts user experience, latency, and resource utilization, making the communication model a critical component of our problem formulation. Our research introduces a dynamic, resilient ES architecture where each BS connects to both a primary and a backup ES. Unlike traditional systems, where one backup ES handles the entire workload of a failed ES, our approach distributes the workload across multiple backup ESs. For example, if an ES managing three BSs fails, the workload is distributed among the backup ESs. These backup ESs may be the same or different for each BS. Instead of relying on a single backup ES to handle all tasks, different ESs may take over the workloads of different BSs. This mechanism ensures service continuity with minimal disruption. Figure 3.2 illustrates this flexibility with a sample scenario of BS assignments to ESs. Our model supports scenarios where multiple BSs are connected to the same ES, such as ES s1, with backup ESs (e.g., s2, s3) actively engaged in resource allocation and task parallelism. If ES s1 fails, the backup ESs seamlessly assume responsibilities for all connected BSs, maintaining network integrity and service availability. This mechanism shows how our system adapts to various network topologies while ensuring fault tolerance and optimizing load distribution.

Building on the robustness of our model's redundancy mechanisms, we now shift our focus to the computational aspects in the next section.

3.1.4 Computation Model

In our research, we make a fundamental assumption regarding the computational capabilities of ESs. Specifically, we assume that each ES is inherently capable of computing the tasks assigned to it efficiently and effectively. This assumption implies that there is no re-

source competition or scarcity at the computation level. The rationale behind this assumption is rooted in our system design and the constraints outlined in the problem formulation. As we assign BSs to ESs, we ensure that these assignments adhere to various constraints, including geographical proximity, capacity constraints, and resiliency requirements. By carefully managing the assignments and verifying that these assignments satisfy these constraints, we guarantee that each ES is well-provisioned with the necessary computational resources to handle the tasks it receives. Consequently, there is no contention for resources among tasks, and each job can seamlessly access the computational power it requires. This assumption simplifies our computation model, allowing us to focus on other critical aspects of latency optimization within the MEC network. It ensures that our analysis and findings are based on a system configuration where computational capacity is not a limiting factor, contributing to the robustness and efficiency of our research. This foundation allows us to effectively assess the impact of computational capabilities on overall system performance, leading into a closer examination of the factors contributing to latency in MEC systems.

3.1.5 Latency Components

In the context of MEC, understanding the factors contributing to the total time latency of an edge service is crucial. This latency can be broken down into several key components, as highlighted in the study on learning-based mobility management under uncertainties for MEC (Wang et al., 2018) [89].

The total time latency, denoted as D^{edge}, encompasses three primary elements:

- **Transmission Latency (D^{trans}):** This component represents the time it takes for data to travel from the source (typically a BS) to the ES. It encompasses factors such as signal propagation, transmission speed, and the distance between the source and the server.

- **Queuing Latency:** In conventional scenarios, queuing latency arises when multiple tasks or requests compete for the same computational resources, leading to delays as they await their turn. However, our research makes a specific assumption that underpins the elimination of queuing latency. We posit that the total workload imposed by BSs on an ES never exceeds its capacity. In other words, we ensure that each ES is meticulously allocated tasks in a manner that prevents resource contention. As a result, queuing latency, which typically emerges from resource conflicts, is effectively

eliminated in our system design. Each task receives the required computational resources promptly and without contention, contributing to the low-latency objectives of our research.

- **Computation Latency (D^{comp}):** This element reflects the time it takes for the ES to process the received data and generate results. Computation latency is influenced by the complexity of the task and the available computational power of the server.

Our research simplifies the latency model by excluding queuing latency, as we assume that each job receives the necessary computational resources without competition. Therefore, the overall time latency for an edge service, denoted as D^{edge}, is defined as the sum of transmission latency and computation latency, as shown in equation (3.1).

$$D^{\text{edge}} = D^{\text{trans}} + D^{\text{comp}} \tag{3.1}$$

It is important to note that, in our model, we specifically consider uplink transmission latency, as we assume that the size of the task result is relatively small compared to the data transmitted to the ES. This understanding of latency components sets the stage for our exploration of resource allocation strategies, where we aim to enhance resilience by optimizing the use of both primary and backup ESs.

3.1.6 Resource Allocation for Enhanced Resilience

In our pursuit of designing a robust and resilient MEC network, we explore an innovative approach that involves resource allocation in both the primary and backup ESs for a given BS. Our objective is to ensure uninterrupted service, particularly for delay-sensitive traffic, even in the face of ES failures. This approach represents a deliberate trade-off between resource utilization and performance, with a focus on enhancing the overall quality of service.

3.1.6.1 Resource Allocation Strategy

In this approach, both the primary and backup ESs designated for a BS are actively involved in the processing of tasks. Traditionally, the primary ES handles the majority of task processing, while the backup ES remains on standby, ready to take over in case of primary ES failure. However, our novel strategy allocates computational resources in both ESs simultaneously. By allocating resources in both ESs, we enable task parallelism which

means that both ESs can work in tandem to calculate tasks and prepare output. This approach is particularly advantageous for delay-sensitive applications, such as AR gaming or autonomous vehicle navigation, where even a momentary interruption in service can lead to a poor user experience. The primary motivation behind this resource allocation strategy is to ensure uninterrupted service, even in the event of a primary ES failure. With both ESs actively engaged, there is virtually no delay in task handover and execution when the primary ES encounters a failure.

3.1.6.2 Trade-off Between Resource and Performance

It is important to acknowledge that this approach requires additional computational resources. Both the primary and backup ESs must be adequately provisioned to handle tasks concurrently. This requirement represents a trade-off between resource allocation and performance optimization. In a MEC network, where the delivery of high-quality services is paramount, this trade-off is justifiable. The benefits of ensuring uninterrupted service, especially for latency-sensitive traffic, far outweigh the additional resource costs.

In summary, our resource allocation strategy, which involves both the primary and backup ESs in task processing, represents a strategic trade-off between resource utilization and performance optimization. By prioritizing uninterrupted service for delay-sensitive traffic, our method aligns with the core objective of MEC networks—delivering high-quality, responsive services to end-users. This trade-off between resource utilization and performance optimization forms the foundation for the challenges we address in the ES placement problem, which we now formalize in our problem formulation.

3.2 Problem Formulation

The challenge of positioning ESs in MEC environments can be visualized as a network, embodied as an undirected graph denoted as $G = (V, E)$. This graph encompasses mobile users, BSs, and potential ES sites. The vertex set V, resulting from the union of B (BSs) and S (potential ES locations), represents all vertices in the network. The edge set E denotes the weighted connections, encompassing links between BSs and ESs (BS-ES) and those among ESs (ES-ES). These links' weights symbolize propagation delays influenced by geographical distances, calculated as shortest path lengths. Notably, this context does not include other types of links, such as those between BSs or between mobile users

and BSs. Now that we have set the stage for our problem formulation, let us explore the assumptions and constraints guiding our approach.

Some Assumptions and Constraints

- There are K ESs to be deployed in K distinct locations, with K as a constant. This assumption reflects a fixed number of ESs available for deployment, a common assumption in many studies to simplify the problem space.

- ESs can share locations with BSs, implying that potential ES locations match the BS set ($S = B$). Specifically, each ES is placed at a location associated with one of the BSs. Given that the number of BSs is $|B|$, the number of available locations for ESs is also $|B|$. Each BS is directly connected to an ES, either at the same site or a different one. While each BS is connected to an ES, not all BSs will have an ES co-located at their site. In other words, every ES is co-located with a BS, but not all BSs have an ES at their site. All mobile user requests from a BS will be processed by an ES.

- Each ES has a finite processing capacity to process client requests and serves a specific number of BSs from set B. It is also assumed that each ES is identical and has the same limited computation resources (c_s) to process mobile user requests. These assumptions simplify the problem by considering uniformity among ESs and their processing capabilities.

- It is assumed that all ESs are placed, meaning that there are no unplaced ESs and no unallocated ESs. All BSs must be assigned to ESs, and each BS is assigned to two ESs. This assumption ensures full coverage and utilization of ES resources.

- BSs can establish connections with multiple ESs to ensure redundancy in case of server failures, thereby enhancing the robustness of the system. This constraint prevents single points of failure and ensures fault tolerance.

- Each BS connects to a primary ES server and one or more backup ESs. If the primary server fails, backup ESs take over. BSs sharing the same primary server may have different backup ESs. This setup ensures that in case of a primary server failure, each BS can still receive service from different backup ESs. This arrangement ensures resilience without relying on a single backup ES for all BSs.

- No BS can have the same ES as both its primary and backup server. This constraint prevents single point of failure and ensures fault tolerance in the system.

The assumptions and constraints outlined in this section lay the groundwork for our problem formulation, providing a clear understanding of the parameters and conditions guiding our ESP model. These assumptions, such as the fixed number of ESs, co-location arrangement with BSs, uniformity among ESs, and full coverage of ES resources, simplify the problem space while ensuring efficient resource utilization and fault tolerance within the system. Moving forward, we will introduce binary variables to represent key decision variables in our optimization model, further refining our approach to ESP in MEC networks.

Binary Variables: y_j represents the ES's location decision. Its value is 1 if BS j's location is selected to deploy an ES, and 0 otherwise. $x_{i,j}$ denotes the assignment of BSs to ESs. If BS b_i is assigned to s_j, then the value of $x_{i,j}$ is 1; otherwise, it is 0. The binary decision variable is used for all BSs and ESs, denoted by $1 \leq i \leq |B|$ and $1 \leq j \leq K$ respectively. Here, $|B|$ denotes the total number of BSs, and K represents the total number of ESs. In the following section, we will develop the objective function, encapsulating the key optimization goals for our ESP model in MEC networks.

Objective Function: The resilient ESP problem in MEC networks is defined with three primary goals: minimizing access delay between BSs and ESs (d_{ij}), ensuring equitable distribution of workload among ESs, and integrating resiliency measures (r). As a result, the placement of ESs within an MEC network becomes a multifaceted minimization challenge. This challenge involves reducing access delays (first component in objective functions 3.4 and 3.12), minimizing workload variance among ESs (second component in objective functions 3.4 and 3.12), and incorporating resiliency (equation 3.20) to ensure robust performance even in the face of potential failures.

Each BS is paired with its corresponding ES, and the access delay to the ES is directly proportional to the distance between the BS and ES. We aim for shorter distances from all BSs to their assigned ESs, ideally placing ESs in denser areas. For each BS, the normalized distance is d_{ij}. For the workload, we prefer a balanced distribution among edge servers, evaluated using the standard deviation. A lower standard deviation indicates a more balanced workload. We use the sum of squared differences for the workload to balance the probability of each ES placement. Initially, we calculate the average workload of each ES

by considering the workload of all associated BSs. Let t_i denote the workload of base station i, and K represent the total number of edge servers.

$$\bar{W} = \frac{\sum_{i=1}^{|B|} t_i}{K} \tag{3.2}$$

Next, we compute $W_{\max}$ using the average workload $\bar{W}$.

$$W_{\max} = (\sum_{i=1}^{|B|} t_i - \bar{W})^2 \tag{3.3}$$

The equation 3.3 represents the squared difference between the total workload of the network (if all BSs are connected to a single ES) and the average workload per ES. Here $\sum_{i=1}^{|B|} t_i$ denotes the total workload of all BSs and $\bar{W}$ is the average workload per ES. This measure indicates the degree of imbalance between the total and the average workloads. The optimization process aims to minimize this imbalance to ensure a fair distribution of workloads among the ESs. To simplify subsequent processing, we normalize both the distance and workload to fall within the range [0, 1]. Finally, we use the weighted sum of the normalized distance and workload as our objective function. Let the variable μ represent the weight assigned to distance and workload components, and d_{ij} represent the access delay (distance) between BS i and ES j. Thus, the ESP task within an MEC network is formulated as a singular objective constraint optimization problem, represented by the following equation.

$$\min_{\{x_{ij}\},\{y_j\}} \{\mu \sum_{j=1}^{|S|} \sum_{i=1}^{|B|} x_{ij} y_j d_{ij} + (1-\mu) \sum_{j=1}^{|S|} y_j \frac{1}{W_{\max}} (\sum_{i=1}^{|B|} x_{ij} t_i - \bar{W})^2\} \tag{3.4}$$

Subject to

$$y_j \geq x_{ij},\ \forall i = 1, \cdots, |B|, \forall j = 1, \cdots, |S| \tag{3.5}$$

$$\sum_{i \in B} t_i x_{ij} \leq c_s, \quad \forall j \in S \tag{3.6}$$

$$\sum_{j=1}^{|S|} x_{ij} \geq r,\ \forall i = 1, \cdots, |B| \tag{3.7}$$

$$\sum_{j=1}^{|S|} y_j = K \tag{3.8}$$

$$d_{ij}x_{ij} \leq bs_{max}, \quad \forall i \in B, \forall j \in S \tag{3.9}$$

$$d_{j`j``}y_{j`}y_{j``} \leq ss_{max}, \quad \forall j`, j`` \in S \tag{3.10}$$

$$x_{ij}, y_j \in \{0, 1\} \tag{3.11}$$

This formulation in equation 3.4, along with the constraints, aims to minimize both the access delay between BSs and their corresponding ESs, as well as the workload variance across all ESs. The constraints for the resilient ESP problem in MEC networks can be summarized as follows: the constraint in (3.5) ensures that a BS is not assigned to an ES if the ES does not exist. This constraint prevents the assignment of a BS to a non-existing ES. The constraint (3.6) guarantees that the total workload of the BSs assigned to an ES does not exceed the capacity of that ES. This constraint ensures that an ES can handle the workload generated by the assigned BSs without surpassing its computation resource capacity (c_s). The constraint (3.7) indicates that each BS is connected to more than one ES, with the resilience factor (r) being greater than 1. This constraint ensures that each BS has backup ES nodes to provide resilience against single ES node failures. However, this work does not consider cases with multi-ES node failures ($r > 2$). The constraint (3.8) ensures that the sum of all binary variables y_j (which equal 1 if an ES is placed at location j and 0 if not) must be equal to K. In simpler term, this constraint guarantees that the total number of deployed ESs across all potential locations exactly matches the target number of ESs (K), as previously defined. In essence, it guarantees that we deploy the specified number of ESs, no more and no less, which is crucial for optimizing the network's performance and resource utilization. The constraint (3.9) ensures that the maximum allowed latency between a BS and its assigned ESs ($bs_{\max}$) is satisfied. This constraint sets an upper bound on the latency between a BS and the ESs (d_{ij}) it is assigned to. The constraint (3.10) specifies that the latency between the ESs ($d_{j`j``}$) should satisfy the latency bound ($ss_{\max}$). This constraint limits the latency among the ESs to a specific maximum value. The constraint (3.11) represents the integrality constraint, indicating that the binary decision variables y_j and x_{ij} should take binary values (0 or 1).

Since equation (3.4) is non-linear and has product of binary variables, we reformulate it by introducing a new binary variable $w_{ij} = x_{ij}y_j$, using the McCormick envelopes [90].

This transformation eliminates the product of binary variables in the objective function, allowing the equation to be rewritten as

$$\min_{\{w_{ij}\}}\{\mu\sum_{j=1}^{|S|}\sum_{i=1}^{|B|}w_{ij}d_{ij}+(1-\mu)\sum_{j=1}^{|S|}\frac{1}{W_{\max}}(\sum_{i=1}^{|B|}w_{ij}t_i-\bar{W})^2\} \tag{3.12}$$

Subject to all the above mentioned constraints (3.5 to 3.11) and

$$w_{ij}\leq y_j,\ \forall j=1,\ldots,|S| \tag{3.13}$$

$$w_{ij}\leq x_{ij},\ \forall i=1,\ldots,|B|,\forall j=1,\ldots,|S| \tag{3.14}$$

$$w_{ij}\geq y_j+x_{ij}-1,\ \forall i=1,\ldots,|B|,\forall j=1,\ldots,|S| \tag{3.15}$$

$$w_{ij}\in\{0,1\},\quad \forall i\in B,\forall j\in S \tag{3.16}$$

The constraint (3.16) represents the integrality constraint, ensuring that the binary decision variable w_{ij} take binary values (0 or 1). The non-linear constraint in (3.10) is linearized as follows [2].

$$w_{j'j''}\geq y_{j'}+y_{j''}-1,\quad \forall j',j''\in S \tag{3.17}$$

$$w_{j'j''}\in\{0,1\},\quad \forall j',\ j''\in S. \tag{3.18}$$

Here, the binary variable $w_{j'j''}$ equals 1 if two ESs are located at nodes j' and j'', respectively, and 0 otherwise. Additionally, the access delay between a BS and an ES is defined by the *Euclidean* distance between them (D). If the lengths of the longest shortest path and the minimum shortest path are $D_{\max}$ and $D_{\min}$, respectively, the following relationship is assumed to hold.

$$D_{\min}\leq bs_{\max}\leq ss_{\max}\leq D_{\max}. \tag{3.19}$$

To enforce the constraint that the primary and backup servers for a BS must be different ESs, we introduce binary variables. Let p_{ij} and q_{ij} be binary decision variables defined as follows.

- $p_{ij}=1$ if BS i is assigned to ES j as the primary server, 0 otherwise.

- $q_{ij} = 1$ if BS i is assigned to ES j as the backup server, 0 otherwise.

To ensure that the primary and backup servers for each BS are different, the following constraints are applied.

$$\sum_j p_{ij} + \sum_j q_{ij} \leq C, \quad \forall i \tag{3.20}$$

Subject to

$$p_{ij} + q_{ij} \leq 1, \quad \forall i, \forall j \tag{3.21}$$

$$\sum_i p_{ij} = 1, \quad \forall j \tag{3.22}$$

$$\sum_j p_{ij} = 1, \quad \forall i \tag{3.23}$$

$$\sum_i q_{ij} \leq 1, \quad \forall j \tag{3.24}$$

$$\sum_j q_{ij} \leq 1, \quad \forall i \tag{3.25}$$

The constraints are interpreted as follows. The *Primary and Backup Server Count Constraint* in equation (3.20) ensures that for each BS i, the sum of p_{ij} (primary servers) and q_{ij} (backup servers) across all possible ESs j must be at most C. This guarantees that each BS is connected to a total of C servers (combining both primary and backup). In this work, C is set to 2, ensuring that each BS has exactly one primary server and one backup server. The *Assignment Constraint* in equation (3.21) enforces that, for each BS i and each ES j, either p_{ij} or q_{ij} (or both) can be 1, but their sum cannot exceed 1. This ensures that each BS is assigned to at most one primary server and one backup server. The *Primary Server Selection* constraint in equation (3.22) guarantees that for each ES j, exactly one BS (i) is assigned to it as the primary server. The *BS Assignment Constraint* in equation (3.23) ensures that each BS i is assigned to exactly one ES as its primary server. The *Backup Server Selection* constraint in equation (3.24) guarantees that for each ES j, at most one BS (i) is assigned to it as the backup server. Finally, the *BS Backup Assignment Constraint* in equation (3.25) ensures that each BS i is assigned to at most one ES as its backup server.

These constraints ensure that each BS has exactly one primary server and one backup server. Consequently, each BS has a distinct primary and backup server, with no ES serving

as both for the same BS. This distinction enforces diversity in server selection, thereby enhancing the resilience of the ESP in MEC networks. The comprehensive formulation optimizes network performance while accounting for resiliency against failures. The proposed solution, detailed in the following section, addresses these challenges with an innovative approach.

Table 3.1 summarizes the notation used in the formulation of the Resilient ESP Problem, and the proposed solution in Section 3.3.

Table 3.1: Notation used in the problem formulation and proposed algorithm.

Symbol	**Definition**
$\|B\|$	Total number of BSs
$\|S\|$	Total number of potential locations where an ES can be placed
B	Set of all BSs in the network
$bs_{\max}$	Latency bound between a BS and its assigned ESs
C	Number of ESs combining both primary and backup for a BS
c_s	Capacity of an ES
d_{ij}	Access delay between BS i and ES j
$d_{j'j''}$	Access delay between ES j' and ES j''
E	Set of links between a BS and an ES at a location in S
G	MEC network (Undirected graph)
K	Total number of ESs
p_{ij}	1 if BS i is assigned to ES j as the primary server, and 0 otherwise
q_{ij}	1 if BS i is assigned to ES j as the backup server, and 0 otherwise
r	Number of ESs to serve a given BS (resilience parameter)
S	Potential ES locations (Set of all ESs that will be placed in the network)
$ss_{\max}$	Inter-server (ES to ES) latency bound
t_i	Workload of each BS i
V	Vertex set
$\bar{W}$	Average workload of ES
W_i	Workload of each ES i.
$w_{j'j''}$	1 if two ESs are at BS j' and j'', respectively, and 0 otherwise
$W_{\max}$	Maximum workload deviation (the maximum difference between the workload of an ES and its average)
$x_{i,j}$	1 if BS i is connected to the ES at location j, and 0 otherwise
y_j	1 if BS j is selected to deploy an ES, and 0 otherwise
μ	The weight assigned to distance and workload components, and it is in the range (0, 1)

3.3 Proposed Solution

In this section, we introduce our novel approach to solving the ESP problem in MEC networks. This complex problem involves the strategic allocation of BSs to ESs with the objectives of minimizing access delay, balancing workloads, and ensuring system resilience. Our solution is implemented through two main steps: 1) determining strategic ES locations and 2) employing a heuristic algorithm for BS allocation. We revisited the concept of facility location in Section 2.3.1 and explored clustering algorithms, specifically highlighting the role of K-Medoids in Section 2.3.2. These concepts provide the foundation for determining strategic ES locations and allocating BSs to ESs. In the following section, we provide a detailed description of our heuristic solution, offering a comprehensive approach to addressing the ESP problem.

3.3.1 Descriptions of the Proposed Algorithm

In response to the challenges outlined earlier in Section 3.2, Algorithm 1 serves as a general framework that encompasses the common steps of our proposed algorithm, namely RESP (heuristic). This algorithm provides a comprehensive overview of the steps involved in the solution and their significance in optimizing ESP. Our Resilient Edge Server Placement (RESP) Algorithm consists of two key steps.

Step 1: Strategic ES Location Determination

Since ES locations are not predefined, we use cluster-based methods, specifically K-Medoids, to identify the best possible positions for the ESs. Our approach focuses on minimizing access delay, achieving workload balance (as described in Equation 3.12), and ensuring resiliency (as described in Equation 3.20). The constraints such as distance (delay), capacity, and connectivity, discussed in Section 3.2, guide this process.

For calculating access delay between BSs and ESs (d_{ij}), we primarily use the *Euclidean distance*. Additionally, we employ the *Haversine formula* to observe trends and determine if different distance calculation methods affect the outcome. Our objective is to identify K suitable locations from a set of potential ones for placing K ESs. We assume that each ES is positioned alongside a BS, resulting in $|B|$ available locations for ES placement, where $|B|$ denotes the number of BSs. However, not every BS will necessarily have an ES at the same location.

Algorithm 1 Resilient Edge Server Placement Algorithm, RESP (heuristic)

procedure EDGESERVERPLACEMENT(B, K, μ, $bs_{\max}$, Resiliency, c_s)
    $K \leftarrow$ Required number of edge servers
    $S \leftarrow$ Set to hold edge server locations
    $B \leftarrow$ Set to hold base station indexes
    Strategic Edge Server Location Determination:
    Initialization:
        Randomly select K initial medoids from base station locations
    Distance Calculation:
        Compute distance matrix between base stations and edge servers
    while not converged **do**
        Assign nearest medoid:
            Assign each base station to its nearest medoid
        Update medoids:
            Update medoids to minimize total distance within clusters
        Check for convergence:
            Compare new and old assignments for edge server locations
    end while
    $S \leftarrow$ Final medoids represent strategic edge server locations
    Heuristic Algorithm for Base Station Allocation:
    for $b \in B$ **do**
        Distance Calculation:
            Calculate distances to all available edge servers
        Load Assessment:
            Determine total workload of the nearest edge server
        Assignment:
            If the workload of the nearest edge server is below the threshold (c_s):
                Assign the base station to it
            Else:
                Find the next nearest edge server with capacity
        Primary and Backup Connections:
            Assign each base station to the nearest and second nearest edge servers
    end for
    Ensure Resiliency:
    for $b \in B$ **do**
        Check if nearest and second nearest edge servers are the same:
            If they are the same, find the next best edge server that satisfies resiliency
    end for
end procedure

K-Medoids Algorithm for ES Location Selection: To efficiently identify the best possible ES locations, we employ the K-Medoids algorithm. This algorithm selects K representa-

tive data points (medoids) to form clusters, ensuring that each data point (BS) is closer to its assigned medoid (ES location) than to other medoids. By partitioning data points into clusters based on their actual distances, the K-Medoids algorithm effectively supports applications such as ES placement.

Step 2: Heuristic Algorithm for BS Allocation

Once ES locations are established, we apply a heuristic algorithm to assign BSs to these fixed ESs, ensuring compliance with capacity (as defined in Equation 3.6), distance (Equation 3.9, 3.10), and resiliency constraints (equation 3.7). This algorithm determines which ES each BS should connect to, aiming to minimize access delay, balance workloads, and ensure resiliency. For each assignment (both primary and backup connections), we evaluate the defined constraints, considering distance and workload. This approach is inspired by existing ESP techniques in MEC [33, 91].

To achieve these goals effectively, we prioritize placing ESs in denser areas and normalize distances using both the *Euclidean distance* metric and the *Haversine formula*. Initially, we compute the *Euclidean distance* matrix between BSs and ESs. Each distance (d) is then normalized to a range of 0 to 1 based on the matrix's minimum and maximum values. Specifically, for each BS, the normalized distance (d_{ij}) is calculated using the formula:

$$d_{ij} = \frac{d - \text{Min}_d}{\text{Max}_d - \text{Min}_d}, \tag{3.26}$$

where Min_d and Max_d represent the minimum and maximum distances across all BSs. This normalization ensures that all d_{ij} values fall within the range [0, 1].

Similarly, workloads are normalized to a range between 0 and 1 based on the W_{max}, $\bar{W}$, minimum and maximum values. Metric values based on the normalized distances and normalized workloads represent the suitability of assigning a BS to a particular ES as defined by Equation 3.27. A weighted sum of normalized distance and workload yields a single index:

$$\text{metric} = \mu \cdot \text{distance} + (1 - \mu) \cdot \text{workload} \tag{3.27}$$

Here, μ determines the weights assigned to distance and workload components. This metric is based on the theoretical framework outlined in Equation 3.12, which formulates the optimization problem. Specifically, this optimization problem aims to minimize a weighted

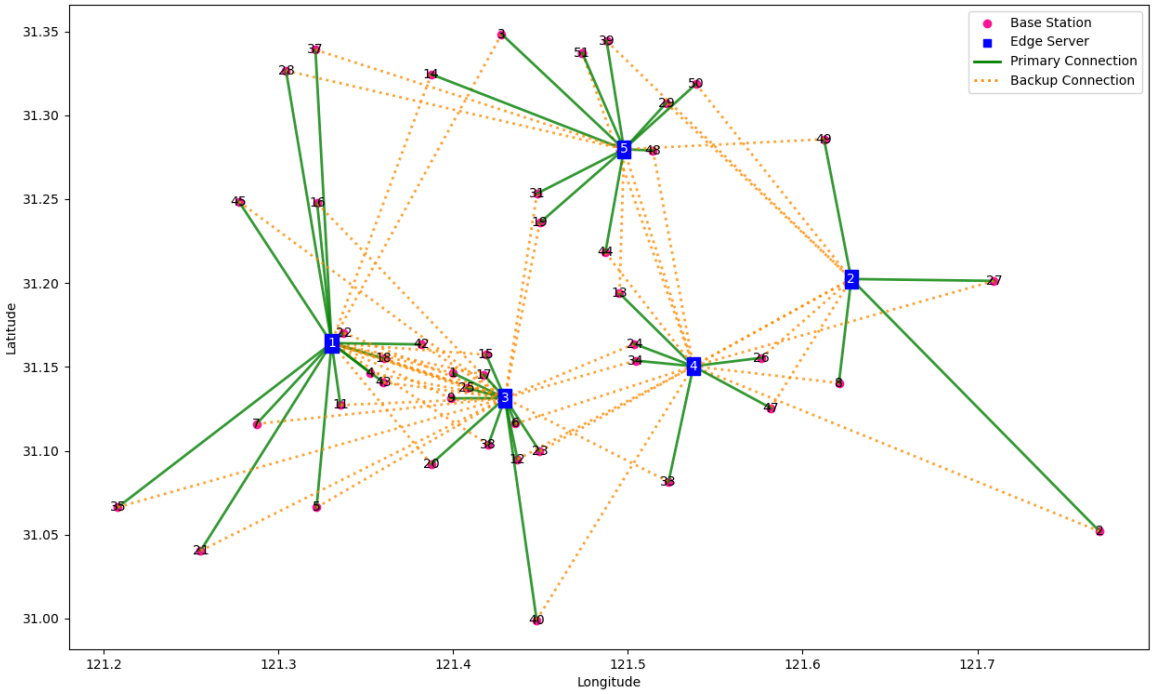


Figure 3.3: ESP with 50 BSs.

sum of distances and workload variance to determine the strategic ES assignment for each BS. We set $\mu = 0.5$ to ensure fairness and equal emphasis on both components. However, this weight is not fixed; it can be varied based on the network's QoS requirements. Each BS is assigned to the nearest ES (primary connection) and the second nearest ES (backup connection) according to the calculated metric values. To ensure workload balance among ESs, we calculate the standard deviation of their workloads, as outlined in Section 4.3.2 and Equation 4.1. This balance guarantees that each mobile device accessing the system receives an equitable share of the computational resources provided by ESs.

Finally, the algorithm ensures resiliency by verifying that each BS has distinct primary and backup ESs. If the nearest and second nearest ESs are the same, the algorithm selects an alternative ES that meets the resiliency criteria. This technique gurantees effective ES placement while considering primary and backup connections and system constraints (as detailed in Equations 3.4 to 3.25. Our two-step strategy ensures a balanced, low-latency, and resilient ESP, effectively addressing the core challenges of MEC networks. Figure 3.3 illustrates the spatial distribution of ESP with 50 BSs, providing a visual representation of the heuristic deployment strategy discussed in this section.

In summary, our proposed solution integrates cluster-based techniques, the K-Medoids approach, and heuristic algorithms to optimize ESP. This approach ensures workload balance, low access delay, and resiliency, making it well-suited for the challenges of MEC network. Having outlined the framework of our proposed algorithm, we now turn our attention to analyzing its time and space complexity. Understanding these metrics will provide insight into the computational resources required to implement our solution effectively and its scalability in real-world scenarios.

3.3.2 Complexity Analysis of the Proposed Algorithm

To analyze the computational and space complexity of the ESP algorithm, let us break down each step and assess its complexity:

1. *Initialization*: Randomly selecting K initial medoids from BS locations has a time complexity of $O(K)$.

2. *Distance Calculation*: Computing the distance matrix between BSs and ESs involves calculating the distance between each BS and each ES. If there are B BSs and K ESs, this step has a time complexity of $O(B \times K)$.

3. *Strategic ES Location Determination (K-Medoids)*:

   - The iterative process of assigning BSs to the nearest medoid, updating medoids, and checking for convergence involves iterating until convergence. The number of iterations depends on the convergence rate, which is algorithm-dependent.
   - Assuming N iterations until convergence, where N is a constant, the time complexity of this step is $O(N \times B)$.

4. *Heuristic Algorithm for BS Allocation*:

   - Calculating distances to all available ESs for each BS has a time complexity of $O(B \times K)$.
   - Assessing the workload and assigning BSs to ESs has a time complexity of $O(B \times K)$.
   - Assigning primary and backup connections also has a time complexity of $O(B \times K)$.

5. *Ensuring Resiliency*:

- Checking if the nearest and second nearest ESs are the same has a time complexity of $O(B)$.

Overall, the computational complexity of the algorithm can be approximated as $O(B \times K)$, where B is the total number of BSs and K is the required number of ESs.

As for space complexity, the algorithm primarily involves storing the distance matrix, the set of ES locations, and BS indexes. Assuming each distance calculation requires constant space and ignoring auxiliary variables, the space complexity is also approximately $O(B \times K)$.

It is important to note that these complexities are based on a high-level analysis and may vary depending on specific implementation details and optimization techniques. Additionally, the computational complexity of the K-medoids algorithm depends on the convergence rate, which can vary depending on factors such as the data distribution and the choice of distance metric. Moreover, implementing redundancy in the system incurs additional costs in terms of hardware, bandwidth, and maintenance. Redundant components ensure that if one component fails, another can seamlessly take over its function, thereby enhancing resilience.

3.4 Conclusion

This chapter addressed the ESP problem by presenting a comprehensive approach aimed at minimizing access delay, balancing workloads, and ensuring resilience in MEC networks. We defined a robust system model, supported by key assumptions and communication/computation models, to tackle the unique challenges of ESP. Our resource allocation strategy was designed to enhance resilience, ensuring system reliability even in the event of server failures. The proposed solution, based on facility location and clustering algorithms, demonstrates the potential for effective ESP optimization. Furthermore, the complexity analysis of our algorithm highlights its computational feasibility and scalability. In the following chapter, we will compare the performance of our heuristic algorithm with established placement solutions. This comparison will provide critical insights into their practical efficiency and effectiveness in real-world MEC deployments.

Chapter 4

Performance Evaluation and Analysis

In this chapter, we provide a detailed description of our experimental setup and assess the performance of our proposed solutions using various methods, metrics, and parameters. To evaluate our approach, we use a real-world dataset from Shanghai Telecom [92]. We implemented our proposed method to verify its performance and compared it with several representative placement strategies in terms of workload balancing, access delay under different ES workloads, and varying numbers of ESs. Extensive experimental evaluations indicate that our proposed heuristic method is both effective and efficient.

4.1 Experiment Setup

We conducted experiments on approximately 80 topologies from the publicly available Shanghai Telecom dataset [92], commonly used in MEC network research [1, 20, 21, 23, 25, 27, 31, 33, 34, 36, 45, 69, 75, 77, 91]. For analysis, we categorized the dataset into two groups based on the number of BS: Group 1 (*small-size*) with $n \leq 200$, Group 2 (*large-size*) with $n > 200$. We selected representative networks from each group to cover diverse topology types.

Our experiment setup rigorously validated our method using real-world BS data in a Python 3.11 environment with Jupyter Notebook. The experimental process was structured into three key phases: First, we conducted a comparative analysis of our approach against existing methodologies to evaluate performance superiority. Next, we determined the ideal number of ESs (K) required for the best results. Finally, we evaluated the impact of varying parameters to test the robustness and adaptability of our method.

Building on our comparative analysis, we designed experiments to assess different performance aspects. Initially, we varied the BS count (n) from 100 to 3000 while adjusting the number of ESs and maintaining a fixed placement ratio ($R = 0.1$). Next, we explored BS counts from 300 to 3000 with a constant number of ESs ($K = 150$). We then examined the impact of varying the ES count (K) from 100 to 500 with a fixed BS count of 3000. Additionally, we investigated how changes in the placement ratio (R) from 0.04 to 0.15 influence performance, with a constant BS count of 3000. We also assessed resilience by randomly deactivating different numbers of ESs. Furthermore, smaller datasets were used to test our approach with BS counts ranging from 20 to 200 and ES counts from 5 to 30 to evaluate performance under various configurations. Throughout all experiments, a consistent weight (μ) of 0.5 was applied to both distance and workload components to ensure fair evaluation and comparison.

All experiments were performed using the same hardware and software configurations, featuring an 11th Gen Intel® Core™ i7-11370H @ 3.30GHz processor, 16 GB of RAM, and a 64-bit operating system with x64-based architecture. Python 3.11 was used for executing the experiments, ensuring consistency and accuracy in our results. With the experimental setup detailed, we will now describe the dataset, including its source and characteristics.

4.2 Dataset Description

Our research leveraged the Shanghai Telecom BS dataset, a comprehensive repository of internet usage data collected from 3233 BSs [92]. This dataset serves as a valuable resource for studying mobile user behavior and network dynamics in urban environments. After meticulous analysis, we identified approximately 3000 active BSs, excluding those with invalid or incomplete data entries. Each record in the dataset contains detailed information about mobile user interactions with BSs, including timestamps for the start and end of communication sessions, as well as the corresponding date and month. Additionally, geographical coordinates (latitude and longitude) are provided for each BS, facilitating spatial analysis and visualization of network coverage shown as in Figure 4.1.

Furthermore, To enhance the dataset's utility, we enriched it with color-coded representations of workload distribution among BSs. This visual aid categorizes BSs into three workload tiers based on the total duration of user interactions:

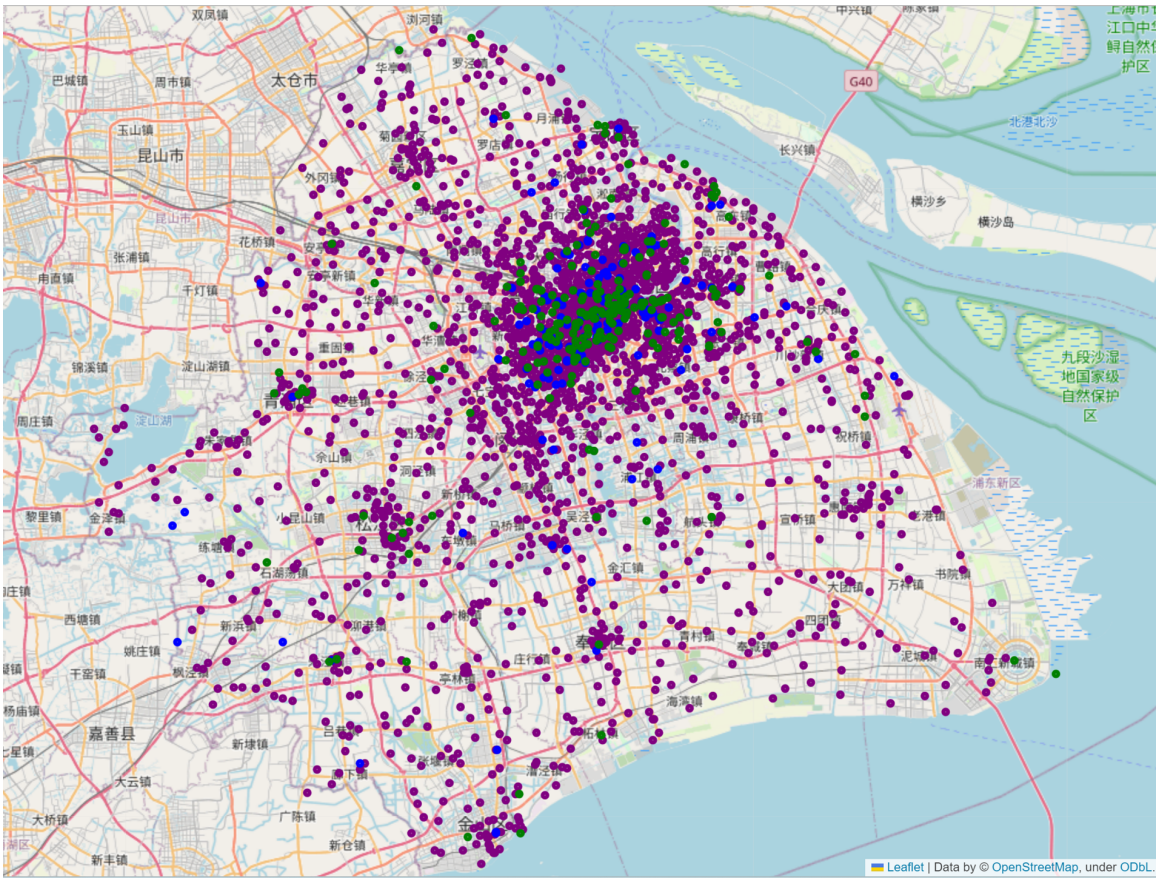


Figure 4.1: Distribution of BSs included in Shanghai Telecom's BS dataset.

- BSs with a workload of less than 200 minutes are represented in green.
- BSs with a workload between 200 and 500 minutes are represented in blue.
- BSs with a workload exceeding 500 minutes are represented in purple.

This color-coded scheme offers a clear visual depiction of workload variations across the network, aiding in the identification of high-traffic areas and network congestion hotspots.

For data analysis purposes, we performed preprocessing steps to filter and standardize the dataset. This involved removing incomplete or erroneous records to ensure data integrity and consistency. Furthermore, we restructured the location information into separate latitude and longitude columns and assigned unique numerical identifiers to each BS for easy reference and analysis. After data preprocessing, we retained 3010 valid data points in the dataset. Some of these BSs are significantly apart from the majority of the BSs, indicating varying geographical distribution.

Table 4.1: Modified base station information.

Base Station ID	Lat	Lon	User number	Workload(min)
1	31.146311	121.399951	3924	188430.2
2	31.051898	121.769904	4359	173749.4167
3	31.348418	121.427796	3365	157120.7833
4	31.146522	121.352501	3828	142505.9
...	...	...	...	...
575	31.197909	121.426542	933	27105.4
1455	31.268106	121.49354	159	6573.65
1905	30.908341	121.870538	17	2037.6
2333	30.852446	121.262447	9	615.15

Using Shanghai Telecom's BS dataset, we represent the workload for each BS as the number of mobile user requests accessing it. Table 4.1 provides a sample of BS information, including workload statistics derived from user interaction timestamps. The workload for each BS, denoted as *Workload (min)*, is measured in minutes and represents the duration of user sessions accessing ESs through that BS. Specifically, this workload is calculated as the difference between the start and end times of mobile user communications for each BS. The corresponding workload for each ES is then determined by aggregating the workloads of all BSs connected to it.

By considering the workload distribution among BSs, we observed a substantial imbalance. Some BSs are heavily overloaded, while others are underutilized or even idle. To optimize mobile application performance, we recognize the urgency of strategically placing ESs to offload the workload from overloaded BSs. This strategic placement of ESs will lead to efficient utilization of edge resources and improved network performance.

In summary, the Shanghai Telecom BS dataset serves as a valuable asset for our research, offering detailed records of mobile user interactions with BSs. By analyzing this dataset, we aim to improve the efficiency and performance of MEC networks, ultimately providing a superior quality of service for end users. With a comprehensive understanding of the dataset, we now turn to the evaluation of our proposed method, comparing its performance with established placement approaches to assess its effectiveness in balancing workloads and minimizing access delay.

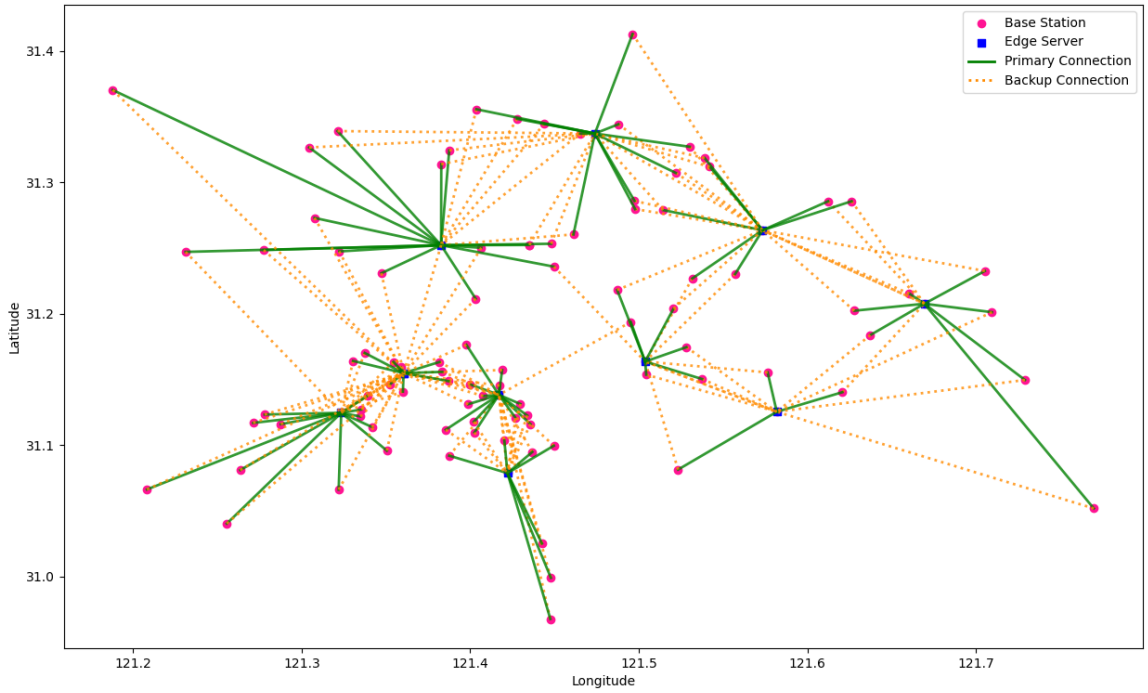


Figure 4.2: ESP with 100 BSs.

4.3 Results and Discussions

In this section, we apply our proposed method to verify its performance by comparing it with several representative placement approaches. Our evaluation focuses on workload balancing, access delay under various ES workloads, and the placement of different numbers of ESs. The Python interface of the Gurobi optimization software (version 10.0.2 (win64)) [93] was used to obtain the optimal solutions. Additionally, Python code was developed to solve the ESP problem considering resilience based on the proposed algorithm. Our experiments were conducted using different dataset sizes, ranging from 15 to 3010 BSs. To visually represent the findings, we plotted all BS coordinates in deep pink circles and marked the ES locations with blue squares. The primary connections of each BS were illustrated using green solid lines, while the backup connections were displayed as dotted orange lines, as illustrated in Figure 4.2 for a sample case with 100 BSs. For each topology, the results demonstrate the feasibility of using RESP (heuristic) to achieve high performance while considering resilience.

In the following section, we first describe the most relevant existing works on the ESP and explain the evaluation metrics. Then, we compare the algorithms using a varying number

of BSs (either with a fixed placement ratio or a fixed number of ESs) and a varying number of ESs. Our experiments cover both small and large datasets. We also present the impact of ES failures on the network with and without resilience. Finally, we provide an analysis of the placement ratio for practical ES deployment scenarios, where determining the appropriate number of ESs can be challenging without a clear reference point. The results from extensive experimental evaluations demonstrate the effectiveness and efficiency of our proposed approach.

4.3.1 Relevant ESP Solutions to Compare

To evaluate the effectiveness of our approach, we compare it with several established placement methods based on workload balance and access delay. These methods include:

1. **Modified K-Means**: The standard *K-Means* algorithm clusters BSs into K groups, with each group centered around an ES [94]. Initially, K-means selects random cluster centers. These centers may not correspond to actual BS locations. The algorithm then iterates to refine these centers. The goal is to minimize the total variance within each cluster, measured as the sum of squared distances between data points and their respective centers [95]. In the *Modified K-Means* approach, we first apply the original K-means to find cluster centers. Then, we map these centers to actual data points in the dataset, specifically the coordinates of BSs or ESs. If the closest BS to a cluster center is already an ES, the next closest BS is chosen as the ES location. This modification ensures that ESs are positioned at actual BS locations. This method results in a more balanced and efficient network, optimizing performance.

2. **Top-K**: Top-K approach involves strategically placing K ESs at the most active BSs, those with the highest number of incoming requests from mobile users. In this method, ESs are positioned at the K busiest BSs, ensuring that the workload is distributed to the most highly accessed locations. Each BS is then assigned to the nearest ES among these selected locations.

3. **Shortest-K**: The Shortest-K approach is designed to optimize ESP by selecting K BSs with the smallest cumulative distance to all other BSs. In this method, ESs are strategically positioned at the K BSs that exhibit the least overall distance from the other BSs. By placing ESs at these minimum distance points, the goal is to minimize access distance for

improved efficiency. Each BS is subsequently matched with the nearest ES among this chosen set of locations, enhancing the network's performance.

4. **Random**: The Random approach involves placing K ESs at BSs in a randomized manner. In this method, K BS locations are selected at random for deploying ESs. Each individual BS is then assigned to the ES that is geographically closest to it. This strategy aims to distribute ESs across the network without a specific pattern, providing potential coverage to a range of locations.

5. **MIP**: Wang et al. [33] frame the ESP problem as a multi-objective constraint optimization challenge. They employ mixed-integer programming (MIP) to determine the optimal placement of ESs, prioritizing workload balancing across servers and minimizing access delay. Experimental validation utilizes datasets comprising approximately 3000 BSs operated by Shanghai Telecom.

6. **MIQP**: Guo et al. [1] introduced MIQP, aiming to optimize ESP by mitigating load imbalance and access delay concurrently. The method first employs K-means to determine ES locations and then formulates the allocation problem as a multi-objective optimization task. To address this, a mixed-integer quadratic programming algorithm is utilized for resolution. Due to MIQP being specifically designed for small datasets, its performance was evaluated on a small dataset with the number of ESs (K) varying from 5 to 30 and the number of BSs (n) ranging from 20 to 200. To maintain clarity and avoid confusion in comparisons with other algorithms (e.g., RESP (heuristic), MIP, Modified K-means, Shortest-K, Top-K, Random) that work on both small and large datasets, the results for MIQP are placed in the Appendix.

Our approach distinguishes itself from the compared methods by formulating an ESP model and utilizing a heuristic algorithm to efficiently allocate BSs to their corresponding ESs. While K-means is widely used in many research studies, K-medoids has been notably underutilized. To achieve a comprehensive evaluation, we have incorporated K-medoids, including its extended version, as one of our experimental techniques.

K-Medoids operates on a similar principle as K-means, dividing BSs into k clusters and assigning them to a centroid [86]. However, a key distinction is in how initial centroids are selected. K-means uses randomly generated centroids, which may not align with actual BS

locations. In contrast, K-medoids employs existing BSs as centroids, eliminating the need for the *determine centroid* step required in K-means. This use of actual BS coordinates simplifies the process. We further enhance the K-medoids approach. In our extended version, K-medoids establishes centroids for the placement of ESs, and a heuristic algorithm is then applied to efficiently allocate BSs to the appropriate ESs. This integration leverages the strengths of K-medoids for centroid determination and the heuristic algorithm for allocation, resulting in improved performance outcomes. With the various placement methods, including our enhanced K-Medoids approach, defined and described, we now turn to the evaluation metrics used to assess their performance.

4.3.2 Evaluation Metrics

In this subsection, we define and elaborate on the metrics used to evaluate the performance of RESP (heuristic) in comparison to various existing ESP methods. The evaluation is based on two key metrics: access delay and workload balancing.

Definition of Access Delay: Access delay is a critical performance metric that measures the time required for communication between BSs and ESs. In our evaluation, we use the average distance between a BS and its associated ES as a representative measure of access delay [1, 33, 91]. While data rates can be used to calculate access delay if available, the lack of public data rates for Shanghai Telecom and the variations specific to operators led us to use access distance as a proxy. Therefore, the distance between a BS and its ES is treated as proportional to access delay, and we use these terms interchangeably.

For distance calculation, we utilized two distinct metrics across our proposed ESP approaches: Euclidean distance and the haversine formula (as discussed and applied in the Appendix). Regardless of the metric used, our algorithm consistently outperformed existing methodologies. To support this claim, we conducted comprehensive analyses using both distance metrics and presented the results, which demonstrate the superior performance of our approach across various scenarios and datasets.

Definition of Workload Balancing (WB): WB refers to the even distribution of computational load among ESs within an edge network. To assess this balance, we employ the standard deviation as a measure [1, 33, 91]. We consider a scenario where K ESs are strategically placed among the BSs. The workload of each ES, denoted as W_i, is calculated, and

subsequently, the standard deviation of these workloads is evaluated using the formula:

$$\text{WB} = \sqrt{\frac{\sum_{i=1}^{K}(W_i - \bar{W})^2}{K}} \tag{4.1}$$

Where:

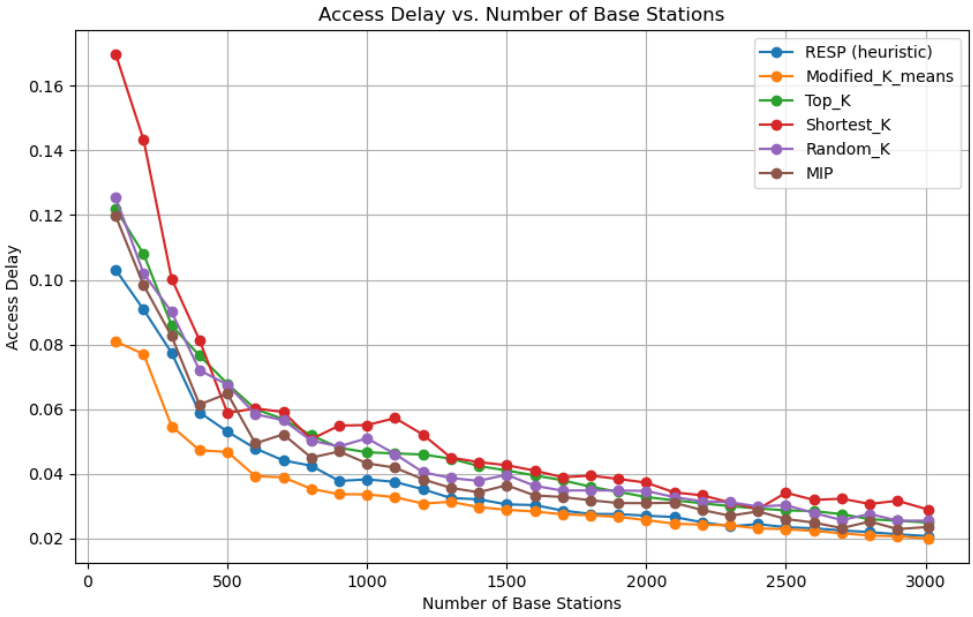
- W_i represents the workload of each ES i.
- $\bar{W}$ denotes the average workload across all ESs.
- K signifies the total number of ESs.

A lower standard deviation indicates a more balanced distribution of workloads among the ESs, reflecting how uniformly the computational load is shared across the network.

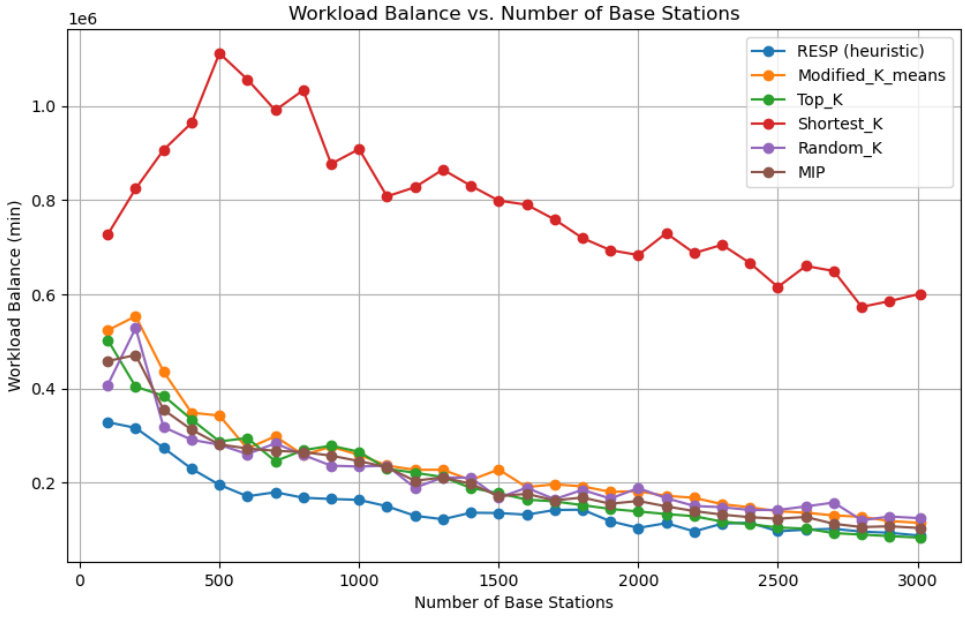
Through these metrics, we provide a comprehensive analysis of our approach compared to existing methods, showcasing its effectiveness in both reducing access delays and achieving balanced workloads. The following sections compare the results of our performance evaluation outcomes, including analyses of each method. We will examine several figures and discuss the impacts of different factors on workload balance and access delay. Specifically, we assess our method by comparing it to other placement strategies in terms of workload balance and access delay. These evaluations provide insights into the scalability and effectiveness of each approach.

4.3.3 Comparison of Results Using Number of BSs

In this section, we conducted a comprehensive performance evaluation using Shanghai Telecom's BS dataset, systematically increasing the number of BSs (n) from 100 to 3000. In Figure 4.3, we use a fixed ES-to-BS ratio (R = 0.1) to evaluate the impact of increasing the number of BSs while maintaining proportional ES placements. This approach helps us understand how well the system scales with a balanced ES-to-BS ratio. Conversely, Figure 4.4 keeps the number of ESs constant (K = 150) to assess how varying BS densities affect performance with a fixed infrastructure. These complementary scenarios provide a comprehensive understanding of how placement strategies perform under different network conditions.



(a) ES access delay



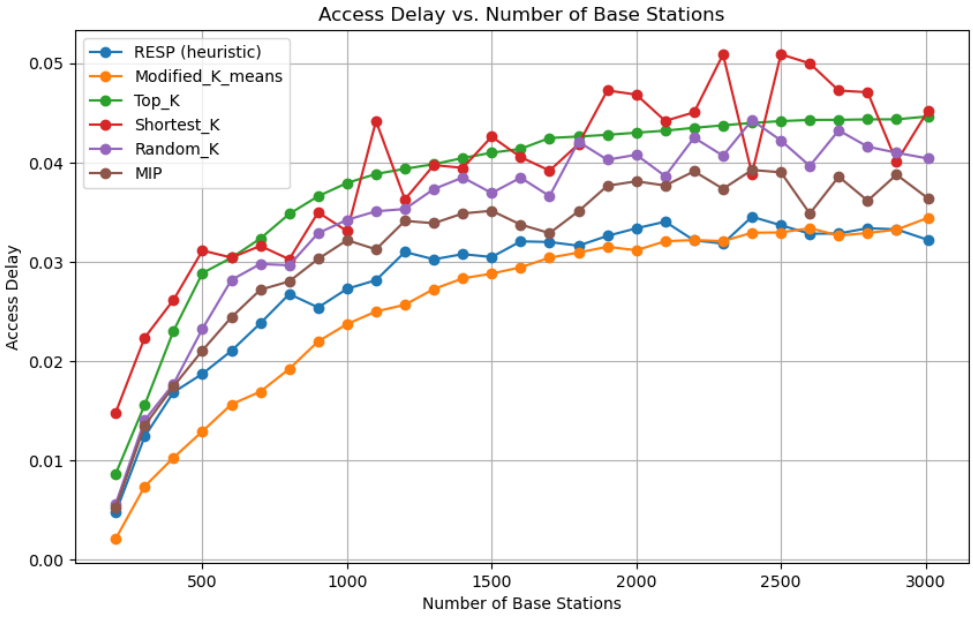
(b) ES workload balancing

Figure 4.3: Comparison of the results with respect to the number of BSs with a fixed number of placement ratio ($R = 0.1$). (a) Access Delay vs. Number of BSs; (b) Workload Balance vs. Number of BSs.

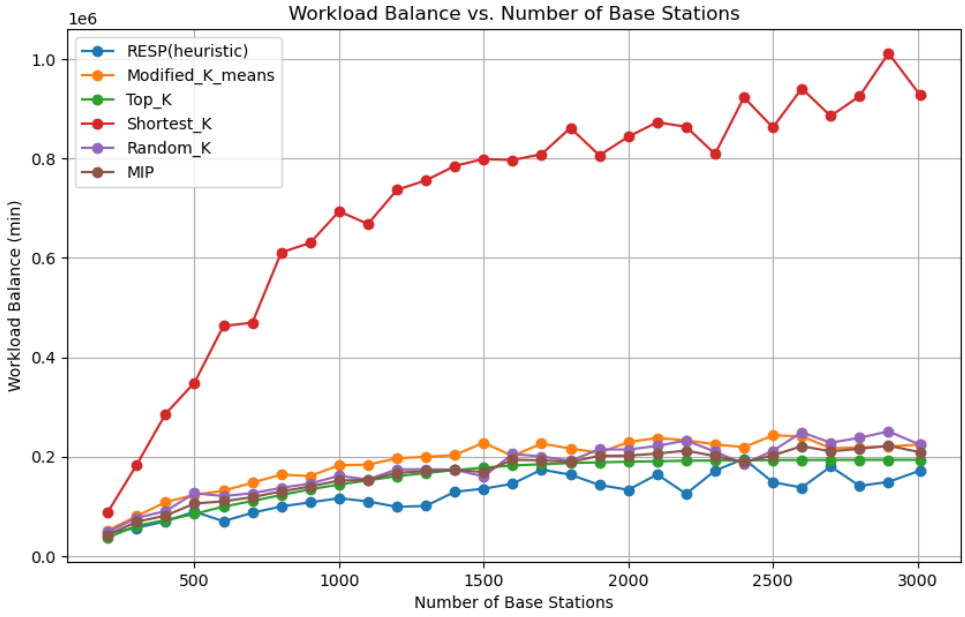
Figure 4.3 illustrates the performance evaluation outcomes across various placement approaches as the number of BSs varies while maintaining a fixed ES-to-BS ratio ($R = K/n = 0.1$). Figures 4.3a and 4.3b represent access delay and workload balancing, respectively. For access delay, the performance ranking is as follows: Modified K-means $>$ RESP (heuristic) $>$ MIP $>$ Random $>$ Top-K $>$ Shortest-K. Modified K-means performs better than RESP (heuristic) in some cases. It starts by selecting cluster centers strategically, though these centers are initially random and may not match with actual BS locations. The method then refines these centers to minimize the sum of squared distances between data points and their centers. This strategy leads to more optimized ES placements in certain scenarios, resulting in lower access delays compared to the heuristic method. For workload balancing, an essential parameter to gauge load distribution, the ranking is as follows: RESP (heuristic) $>$ Top-K $>$ MIP $>$ Random $>$ Modified K-means $>$ Shortest-K. Lower values in workload balance indicate a more evenly distributed load across ESs, which is ideal for network performance. RESP (heuristic) achieves the best workload balance, demonstrating its effectiveness in distributing the workload evenly among ESs.

The results are influenced by several factors. In Shanghai's dense urban environment, the spatial distribution of BSs affects the performance of placement methods. The RESP (heuristic) method excels by anticipating high-demand areas, leading to improved workload distribution. Modified K-means performs better in access delay due to its optimization of ES placement relative to clustered BSs. Interestingly, the Random approach outperforms Top-K in access delay, likely due to the suitability of Random technique for Shanghai's dense urban layout. From the data presented in Figure 4.3, after analyzing the contributing factors and the performance trends, it is evident that our proposed approach consistently outperforms the Modified K-means, MIP, Top-K, Shortest-K, and Random placement methods.

Figure 4.4a and Figure 4.4b provide a comprehensive comparison of results in terms of access delay and workload balance. These results were obtained across BSs ranging from 200 to 3000, while maintaining a consistent number of ESs (K = 150).



(a) ES access delay



(b) ES workload balancing

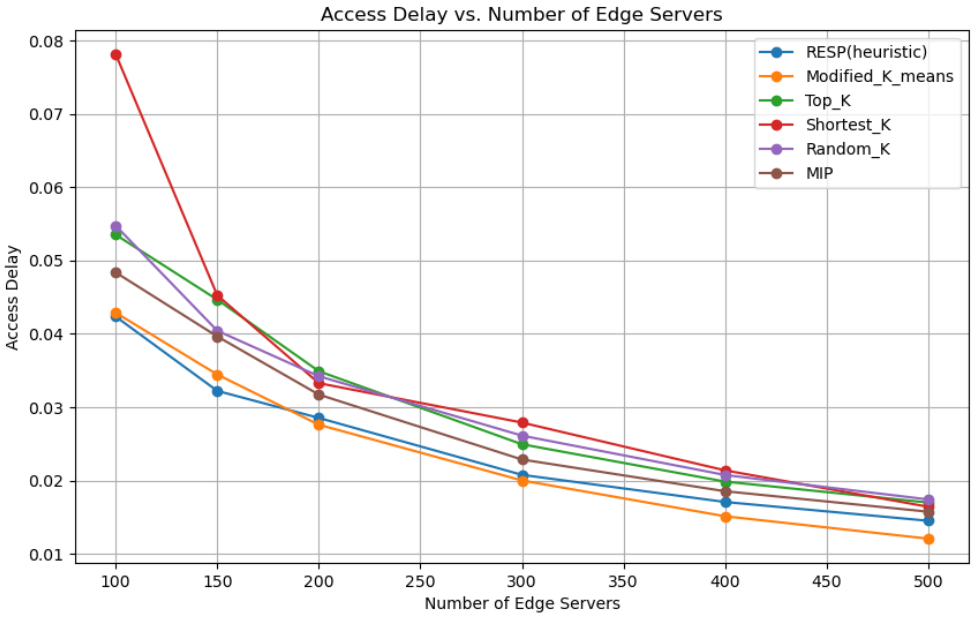
Figure 4.4: Performance evaluation of different approaches with a fixed number of ESs (K = 150) as the number of BSs increases. (a) Access Delay vs. Number of BSs; (b) Workload Balance vs. Number of BSs.

Figure 4.4 shows that the Modified K-means approach exhibits a relatively low access delay. Notably, our approach consistently outperforms Top-K, MIP, Shortest-K, and Random in terms of access delay, with performance nearly on par with Modified K-means. As the number of BSs increases, both access delay and workload balance gradually rise with a fixed number of ESs (K = 150). This trend is attributed to the larger number of BSs served by each ES, leading to greater disparities in workload among these ESs. When considering workload balance, our method consistently outperforms Top-K, MIP, Random, Modified K-Means, and Shortest-K. As in Figure 4.3b, the Shortest-K value in Figure 4.4b is significantly higher than other curves. This approach selects K BSs with the shortest cumulative distances to all other BSs, positioning ESs at these points to minimize access distances and enhance network efficiency. However, when many BSs connect to a limited number of ESs, it can result in increased access delays and workload imbalances, as seen in Figure 4.4. The higher curve value reflects this inefficiency, indicating that while Shortest-K reduces distances, it may fail to distribute loads effectively under high BS counts.

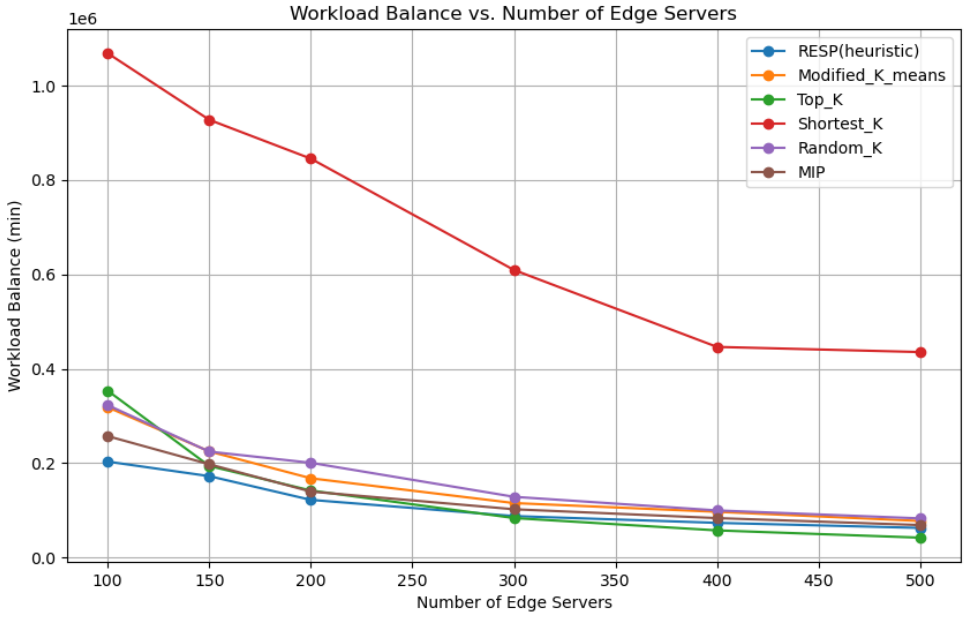
In summary, although our algorithm may not consistently achieve the best results in access delay and workload balance individually, its overall performance surpasses other techniques across different BS counts. Thus, our solution emerges as a favorable choice for ESP optimization. In the following subsection, we explore how the number of ESs impacts access delay and workload balancing, providing a comprehensive analysis to further refine our optimization strategies.

4.3.4 Comparison of Results with Number of ESs

In the preceding subsection, we evaluated the performance of different ESP approaches across varying numbers of BSs. Now, we shift our focus to exploring the impact of varying the number of ESs, denoted as K, on the ES's performance in terms of access delay and workload balancing. Our experiments were based on data from 3000 BSs, and we adjusted K within the range of 100 to 500.



(a) ES access delay



(b) ES workload balancing

Figure 4.5: Impact of the number of ESs on the performance of different approaches. (a) Access Delay vs. Number of ESs; (b) Workload Balance vs. Number of ESs.

Figures 4.5a and 4.5b depict the access delay and workload balancing curves as the number

of ESs, K, increases. Notably, the access delay for all algorithms decreases with an increase in K, as more BSs have the opportunity to be assigned to their nearest ESs. Importantly, our technique outperforms Top-K, MIP, Shortest-K, and Random in access delay. This success is due to effectively balancing ES placement, ensuring BSs are assigned to the nearest available ESs that meet capacity and distance constraints. This balance leads to reduced access delays across the network. Our method closely matches the performance of Modified K-means. Both approaches refine ES placement relative to BS clusters, achieving similarly efficient results in scenarios with densely located BSs.

Figure 4.5b reveals that, as the number of ESs increases, workload balance decreases rapidly. Here, our method excels in workload balancing when compared to MIP, Random, Shortest-K, and Modified K-means. While it does not surpass Top-K in workload balancing, overall, our RESP (heuristic) approach exhibits superior performance as K increases. These findings in Figure 4.5 guide us towards determining an appropriate K while considering specific constraints.

In summary, RESP (heuristic) has been compared with other known placement approaches based on six different performance criteria:

1. Access Delay vs. Number of BSs with a fixed number of Placement Ratio (R = 0.1).
2. Workload Balance vs. Number of BSs with a fixed number of Placement Ratio (R = 0.1).
3. Access Delay vs. Number of BSs with a fixed number of ESs (K = 150)
4. Workload Balance vs. Number of BSs with a fixed number of ESs (K = 150).
5. Access Delay vs. Number of ESs with a fixed number of BSs (n = 3000).
6. Workload Balance vs. Number of ESs with a fixed number of BSs (n = 3000).

Table 4.2 summarizes the overall percentage improvement in access delay and workload balance of RESP (heuristic) compared to other established placement strategies (Modified K-means, Top-K, Shortest-K, Random, and MIP) across all scenarios, covering the six performance criteria mentioned above. Positive values show that RESP (heuristic) outperforms the compared strategy, while negative values indicate that RESP (heuristic) performs worse than the compared strategy.

Table 4.2: Performance comparison (%).

Performance Criteria	Modified_K_means	Top_K	Shortest_K	Random_K	MIP
1	-10.43	19.71	27.13	18.61	10.64
2	33.91	20.12	81.02	29.43	25.43
3	-18.52	26.08	28.33	18.25	10.81
4	33.20	17.66	80.56	27.70	23.21
5	-5.43	18.72	24.37	19.08	11.02
6	25.69	-2.46	83.82	30.37	13.59

Additionally, Table 4.3 presents the performance ranking of all methods based on their overall improvement. For access delay, Modified K-means ranks highest, followed by RESP (heuristic), MIP, Random-K, Top-K, and Shortest-K. This ranking is consistent across most scenarios. For workload balance, RESP (heuristic) typically performs best, except in the final scenario where Top-K shows a slight improvement.

Table 4.3: Performance ranking.

Performance Criteria	Ranking
1	Modified_K_means > RESP (heuristic) > MIP > Random_K > Top_K > Shortest_K
2	RESP (heuristic) > Top_K > MIP > Random_K > Modified_K_means > Shortest_K
3	Modified_K_means > RESP (heuristic) > MIP > Random_K > Top_K > Shortest_K
4	RESP (heuristic) > Top_K > MIP > Random_K > Modified_K_means > Shortest_K
5	Modified_K_means > RESP (heuristic) > MIP > Top_K > Random_K > Shortest_K
6	Top_K > RESP (heuristic) > MIP > Modified_K_means > Random_K > Shortest_K

These comparisons demonstrate that RESP (heuristic) consistently outperforms other existing placement approaches in terms of both access delay and workload balance, indicating its effectiveness in enhancing the performance and resilience of ESP in real-world scenarios. Building on these findings, the next section explores the role of resilience mechanism, comparing the performance of networks with and without resilience to further validate our approach.

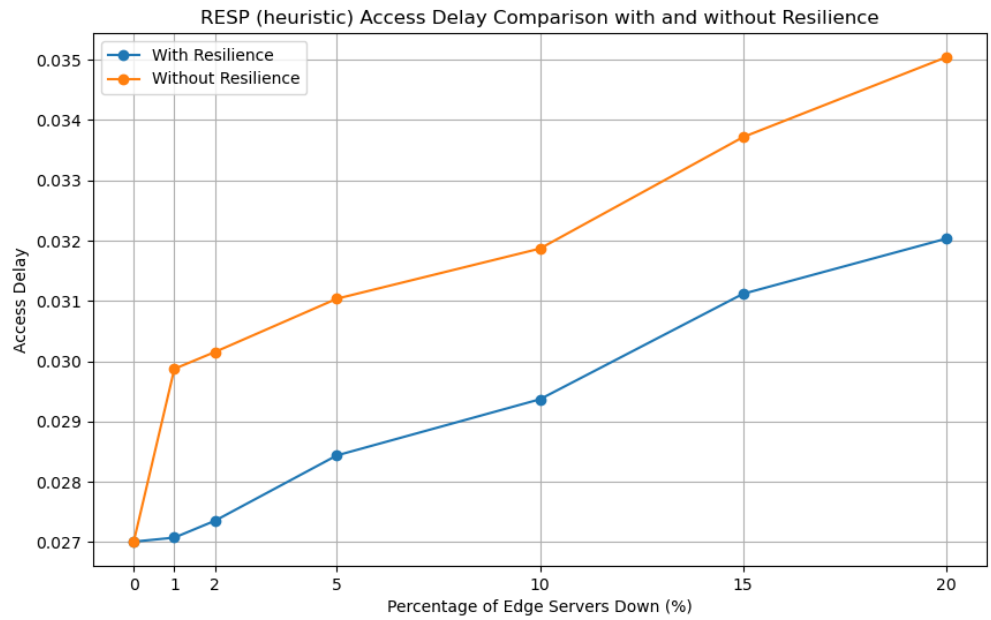
4.3.5 Impact of Resilience

In this section, we assess the impact of resilience mechanisms on network performance. The experiment involved a network of 2000 BSs and 200 ESs, where we randomly deactivated or disconnected varying percentages (1%, 2%, 5%, 10%, 15%, 20%) of the ESs. We monitored the network's behavior under two conditions: with and without resilience features, and subsequently analyzed the performance data. Figure 4.6 illustrates how resilience mechanisms influence network access delay and workload balance as the percentage of ESs down increases. The *With Resilience* condition includes resilience mechanisms, where hot ESs serve as backups for disaster recovery, while the *Without Resilience* condition represents scenarios where no resilience mechanisms are in place. In the 0% server down case, where all ESs are operational, the performance metrics (access delay and workload balance) are identical for both resilient and non-resilient networks. Since resilience mechanisms are designed to manage disruptions, they do not affect the network when no ES failures occur.

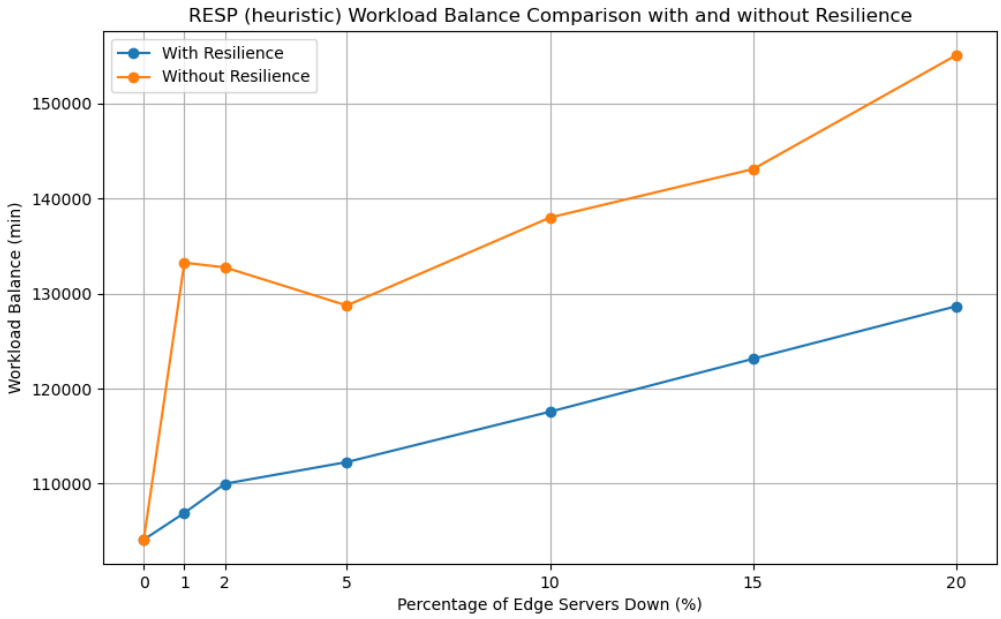
Access Delay Analysis: Figure 4.6a shows the percentage of ESs down on the x-axis and access delay on the y-axis. With resilience mechanisms, each BS has a pre-assigned backup ES, allowing for immediate switch-over and significantly reducing access delay. In contrast, access delay is higher without resilience mechanisms because the network must locate and allocate a backup ES when an ES fails, resulting in increased delay.

Workload Balance Analysis: Figure 4.6b shows the percentage of ESs down on the x-axis and workload balance on the y-axis. With resilience measures in place, workload balance improves even as the percentage of ES failures increases. Conversely, in the absence of resilience mechanisms, workload balance deteriorates. The widening gap between resilient and non-resilient systems as ES failures rise highlights the effectiveness of resilience mechanisms in mitigating the impact of ES failures on workload balance.

The results highlight the critical role of resilience in maintaining network stability during ES failures. However, optimizing the number of ESs is also crucial to ensuring efficient network performance, paving the way for further analysis of critical parameters such as the placement ratio (R).



(a) RESP (heuristic) access delay



(b) RESP (heuristic) workload balancing

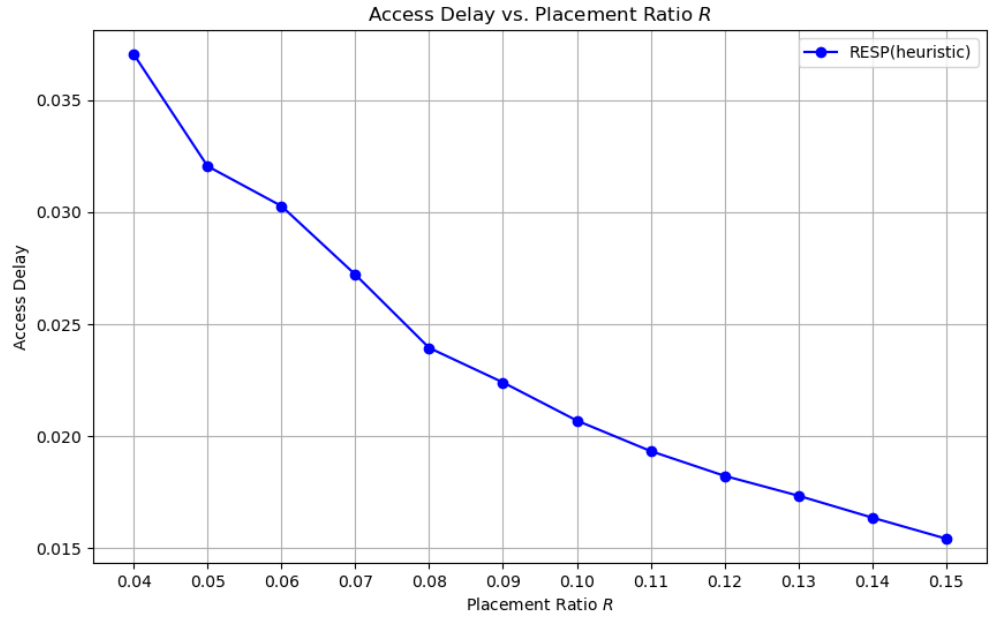
Figure 4.6: Impact of resilience. (a) Access Delay vs. Percentage of ESs Down; (b) Workload Balance vs. Percentage of ESs Down.

4.3.6 Study of Placement Ratio (R)

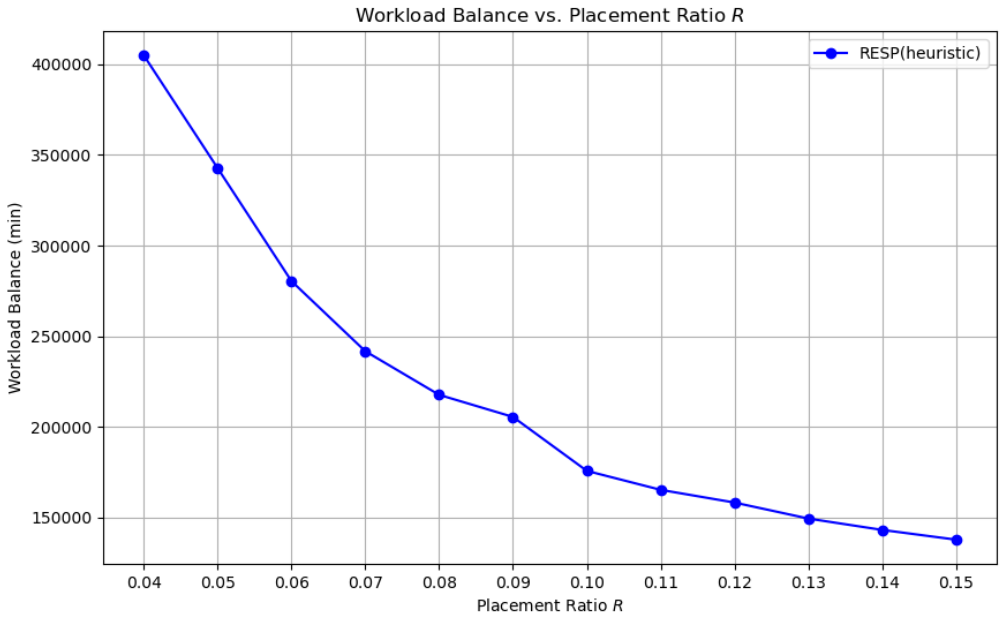
In this section, we examine the placement ratio (R) and its effect on ES deployment efficiency. The placement ratio is a crucial parameter that determines the number of ESs in a network, which directly affects overall network performance. Analyzing this ratio is essential for practical ES deployment, as finding the optimal number of ESs can be challenging without a clear reference point. In our experiments, we varied R from 0.04 to 0.15. Here, R denotes the ratio of the number of ESs to the number of BSs. For example, if R is 0.1 and there are 3000 BSs (n), then K (the number of ESs) would be 300, meaning each ES serves 10 BSs.

Figure 4.7 illustrates the outcomes of our RESP (heuristic) algorithm as we increment the placement ratio R. One noteworthy observation is that the ES access delay reduces as R increases, as a higher R implies more opportunities for each BS to be assigned to its nearest ES, effectively reducing access delays. Furthermore, the workload balance among ESs improves with higher R values. As BS workloads are distributed across more ESs, the disparities in workload size decrease.

As displayed in Figure 4.7, both workload balance and access delay exhibit reductions with increasing placement ratio R. This trend is primarily due to the higher R values allowing for a greater number of ESs to share the BS tasks. Consequently, BSs have a wider selection of ESs to offload their tasks to, resulting in improved performance. This study has direct implications for practical deployment decisions in contexts such as smart cities, which utilize advanced technologies to improve the quality of life for residents. This study helps stakeholders in these environments make informed choices regarding ESP. Factors like infrastructure budget, desired access delay levels, and workload balance can be considered when determining the appropriate value for the placement ratio R.



(a) ES access delay



(b) ES workload balancing

Figure 4.7: Variation in the RESP (heuristic) performance as the parameter R increases. (a) Access Delay vs. Placement Ratios; (b) Workload Balance vs. Placement Ratios.

4.4 Conclusion

This chapter provided a comprehensive evaluation of our proposed solutions, focusing on various metrics, including workload balance, access delay, and the impact of resilience in MEC networks. The results, derived from extensive experiments using a real-world dataset, demonstrate the effectiveness and efficiency of our heuristic method compared to other placement strategies (Top-K, MIP, Random, Modified K-means, and Shortest-K). These findings validate the practical applicability of our approach, offering valuable insights for stakeholders in real-world deployment scenarios. In the following chapter, we will summarize the key findings of this study, draw conclusions, and discuss potential avenues for future work.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

In this thesis, we addressed the critical challenge of optimizing ESP in a MEC environment and proposed the RESP (heuristic) algorithm, a novel approach that leverages the strengths of a clustering-based method and a heuristic algorithm. Our primary objectives were to minimize access delay, balance workload distribution among ESs, and ensure network resilience. By conducting extensive experiments using real data from Shanghai Telecom's BS dataset, we evaluated the performance of RESP (heuristic) alongside other representative methods (e.g., Modified K-means, Shortest-K, Top-K, Random, MIP, and MIQP). Our experimental evaluations have demonstrated the effectiveness of our approach in improving network performance such as reducing access delay and workload variance, and enhancing MEC network resilience, ensuring uninterrupted services.

Our findings highlight that RESP (heuristic) surpasses existing approaches in addressing access delay concerns, particularly within the MEC landscape. Minimizing latency is crucial in ensuring efficient and responsive MEC services, where timely access to computational resources is essential for delivering low-latency applications. Our approach offers an effective means of optimizing ESP to minimize access delay, thus improving the quality of service in MEC networks.

Our research emphasizes the importance of thoughtful ESP, considering factors such as workload balancing in MEC environments. Efficient workload distribution among ESs is vital for optimizing resource utilization and ensuring equitable processing across the net-

work. By addressing workload balancing concerns, our proposed RESP (heuristic) algorithm provides a viable solution for deploying ESs in real-world scenarios. This improvement facilitates the ongoing advancement of MEC technology by enhancing the efficiency and responsiveness of MEC services.

The contributions of this thesis extend beyond the immediate problem of ESP in MEC networks. Our research has shed light on the importance of considering resilience as a fundamental aspect of network design and operation. By integrating resilience considerations into ESP strategies, we have advanced the field of MEC and played a role in the development of more robust and efficient network architectures. Overall, this thesis has presented a comprehensive study on ESP in MEC networks, focusing on resilience.

The findings of this research contribute to the progress of MEC networks and lay the groundwork for future developments in the field of network architecture and design. This research offers valuable insights for decision-makers in deploying ESs in real-world scenarios. It is our hope that this work will inspire further research and innovation in the area of ESP and network resilience, ultimately leading to more efficient and reliable communication networks that meet the ever-increasing demands of users and applications in the digital age.

5.2 Future Work

The success of our resilient ESP approach has unveiled promising directions for future research in the dynamic realm of MEC. Here are several exciting avenues that warrant exploration:

1. *Dynamic Network Scenarios*: Investigate the behavior of our ESP strategy under varying network conditions. This includes scenarios with fluctuating user demand and dynamic ES capacities. Understanding how the system adapts to such real-world fluctuations will be crucial for improving its responsiveness and efficiency.

2. *Scalability*: Address the scalability of our proposed algorithm to cater to larger and more complex networks. As MEC networks expand, accommodating a growing number of ESs and BSs is paramount. Therefore, we will investigate methods for ensuring the algorithm's effectiveness and efficiency at scale.

3. *Holistic Optimization*: Further refine and expand the existing optimization strategies to encompass additional dimensions such as energy efficiency and cost-effectiveness. Our current work has already addressed the importance of preventing underutilization in some servers while avoiding overloading others, thereby enhancing overall server utilization and energy efficiency within the MEC network. However, future research endeavors can further explore advanced techniques and methodologies to optimize energy efficiency and cost-effectiveness even further, ensuring the long-term sustainability and affordability of MEC networks.

4. *Machine Learning and AI Integration*: The integration of machine learning and artificial intelligence techniques holds great potential for refining ESP schemes. These technologies can offer dynamic insights and adaptability, ensuring that ESP remains optimized in rapidly changing environments.

In summary, our current work represents a significant step towards resilient ESP in MEC networks. However, the field of MEC is dynamic and evolving. The future research directions discussed above offer an exciting path forward to enhance the efficiency, adaptability, and sustainability of MEC networks. The combination of cutting-edge technology and comprehensive research will play a pivotal role in shaping the future of this vital domain.

Bibliography

[1] Y. Guo, S. Wang, A. Zhou, J. Xu, J. Yuan, and C.-H. Hsu, "User allocation-aware edge cloud placement in mobile edge computing," *Software: Practice and Experience*, vol. 50, no. 5, pp. 489–502, 2020.

[2] M. Tanha, D. Sajjadi, R. Ruby, and J. Pan, "Capacity-aware and delay-guaranteed resilient controller placement for software-defined wans," *IEEE Transactions on Network and Service Management*, vol. 15, no. 3, pp. 991–1005, 2018.

[3] "ETSI: Multi-access Edge Computing," https://www.etsi.org/technologies/multi-access-edge-computing, accessed: 2024-10-21.

[4] "Hewlett Packard Enterprise - Mobile Edge Computing," https://www.hpe.com/us/en/what-is/mobile-edge-computing.html, accessed: 2024-10-21.

[5] "The Enterprisers Project - What is Mobile Edge Computing (MEC)?" https://enterprisersproject.com/article/2021/2/what-mobile-edge-computing-mec, accessed: 2024-10-21.

[6] M. Liyanage, P. Porambage, A. Y. Ding, and A. Kalla, "Driving forces for multi-access edge computing (mec) iot integration in 5g," *ICT Express*, vol. 7, no. 2, pp. 127–137, 2021.

[7] "ETSI - Mobile Edge Computing: Introductory Technical White Paper," https://portal.etsi.org/portals/0/tbpages/mec/docs/mobile-edge_computing_-_introductory_technical_white_paper_v1%2018-09-14.pdf, accessed: 2024-10-21.

[8] X. Zhang and S. Debroy, "Resource management in mobile edge computing: A comprehensive survey," *ACM Computing Surveys*, 2023.

[9] S. Shahzadi, M. Iqbal, T. Dagiuklas, and Z. U. Qayyum, "Multi-access edge computing: open issues, challenges and future perspectives," *Journal of Cloud Computing*, vol. 6, pp. 1–13, 2017.

[10] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE communications surveys & tutorials*, vol. 19, no. 3, pp. 1628–1656, 2017.

[11] B. Teja Sree, G. Varma, and H. Indukurib, "Mobile edge computing architecture challenges, applications, and future directions," *International Journal of Grid and High Performance Computing (IJGHPC)*, vol. 15, no. 2, pp. 1–23, 2023.

[12] "ScienceDirect - Multi-access Edge Computing," https://www.sciencedirect.com/topics/computer-science/multi-access-edge-computing, accessed: 2024-10-21.

[13] L. Qin, H. Lu, and F. Wu, "When the user-centric network meets mobile edge computing: Challenges and optimization," *IEEE Communications Magazine*, vol. 61, no. 1, pp. 114–120, 2022.

[14] "MongoDB Blog - Computing in the Real World," https://www.mongodb.com/blog/post/computing-in-real-world, accessed: 2024-10-21.

[15] Z. Ning, P. Dong, X. Wang, X. Hu, L. Guo, B. Hu, Y. Guo, T. Qiu, and R. Y. Kwok, "Mobile edge computing enabled 5g health monitoring for internet of medical things: A decentralized game theoretic approach," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 2, pp. 463–478, 2020.

[16] J. Ren, Y. He, G. Huang, G. Yu, Y. Cai, and Z. Zhang, "An edge-computing based architecture for mobile augmented reality," *IEEE Network*, vol. 33, no. 4, pp. 162–169, 2019.

[17] "Simplilearn - Edge Computing vs Cloud Computing," https://www.simplilearn.com/edge-computing-vs-cloud-computing-article, accessed: 2024-10-21.

[18] Z. Wang, Y. Zhou, X. Jin, Y. Chen, and C. Lu, "An edge server deployment approach for delay reduction and reliability enhancement in the industrial internet," *Wireless Networks*, pp. 1–15, 2023.

[19] C. Jian, G. Yan, L. Cheng, Y. Muchuan, L. Jiayu, and J. He, "An optimized cloud edge server placement method based on k-mean algorithm," in *Third International*

Conference on Artificial Intelligence and Computer Engineering (ICAICE 2022), vol. 12610. SPIE, 2023, pp. 1288–1293.

[20] X. Jiang, P. Hou, H. Zhu, B. Li, Z. Wang, and H. Ding, "Dynamic and intelligent edge server placement based on deep reinforcement learning in mobile edge computing," *Ad Hoc Networks*, vol. 145, p. 103172, 2023.

[21] F. Luo, S. Zheng, W. Ding, J. Fuentes, and Y. Li, "An edge server placement method based on reinforcement learning," *Entropy*, vol. 24, no. 3, p. 317, 2022.

[22] W. Li, J. Chen, Y. Li, Z. Wen, J. Peng, and X. Wu, "Mobile edge server deployment towards task offloading in mobile edge computing: A clustering approach," *Mobile Networks and Applications*, vol. 27, no. 4, pp. 1476–1489, 2022.

[23] Z. Hu, X. Xu, and J. Chen, "An edge server placement algorithm based on genetic algorithm," in *Proceedings of the ACM Turing Award Celebration Conference-China*, 2021, pp. 92–97.

[24] X. Chen, W. Liu, J. Chen, and J. Zhou, "An edge server placement algorithm in edge computing environment," in *2020 12th International Conference on Advanced Infocomm Technology (ICAIT)*. IEEE, 2020, pp. 85–89.

[25] K. Cao, L. Li, Y. Cui, T. Wei, and S. Hu, "Exploring placement of heterogeneous edge servers for response time minimization in mobile edge-cloud computing," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 1, pp. 494–503, 2020.

[26] Y. Gong, "Optimal edge server and service placement in mobile edge computing," in *2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, vol. 9. IEEE, 2020, pp. 688–691.

[27] Y. Qu, L. Wang, H. Dai, W. Wang, C. Dong, F. Wu, and S. Guo, "Server placement for edge computing: a robust submodular maximization approach," *IEEE Transactions on Mobile Computing*, 2021.

[28] G. Cui, Q. He, F. Chen, H. Jin, and Y. Yang, "Trading off between user coverage and network robustness for edge server placement," *IEEE Transactions on Cloud Computing*, vol. 10, no. 3, pp. 2178–2189, 2020.

[29] H. Yin, X. Zhang, H. H. Liu, Y. Luo, C. Tian, S. Zhao, and F. Li, "Edge provisioning with flexible server placement," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 4, pp. 1031–1045, 2016.

[30] T. Lähderanta, T. Leppänen, L. Ruha, L. Lovén, E. Harjula, M. Ylianttila, J. Riekki, and M. J. Sillanpää, "Edge computing server placement with capacitated location allocation," *Journal of Parallel and Distributed Computing*, vol. 153, pp. 130–149, 2021.

[31] B. Bahrami, M. R. Khayyambashi, and S. Mirjalili, "Edge server placement problem in multi-access edge computing environment: models, techniques, and applications," *Cluster Computing*, pp. 1–26, 2023.

[32] Y. Shao, Z. Shen, S. Gong, and H. Huang, "Cost-aware placement optimization of edge servers for iot services in wireless metropolitan area networks," *Wireless Communications and Mobile Computing*, vol. 2022, 2022.

[33] S. Wang, Y. Zhao, J. Xu, J. Yuan, and C.-H. Hsu, "Edge server placement in mobile edge computing," *Journal of Parallel and Distributed Computing*, vol. 127, pp. 160–168, 2019.

[34] X. Zhao, Y. Zeng, H. Ding, B. Li, and Z. Yang, "Optimize the placement of edge server between workload balancing and system delay in smart city," *Peer-to-Peer Networking and Applications*, vol. 14, pp. 3778–3792, 2021.

[35] Z. Wang, W. Zhang, X. Jin, Y. Huang, and C. Lu, "An optimal edge server placement approach for cost reduction and load balancing in intelligent manufacturing," *The Journal of Supercomputing*, vol. 78, no. 3, pp. 4032–4056, 2022.

[36] S. K. Kasi, M. K. Kasi, K. Ali, M. Raza, H. Afzal, A. Lasebae, B. Naeem, S. Ul Islam, and J. J. Rodrigues, "Heuristic edge server placement in industrial internet of things and cellular networks," *IEEE Internet of Things Journal*, vol. 8, no. 13, pp. 10 308–10 317, 2020.

[37] B. Shen, X. Xu, L. Qi, X. Zhang, and G. Srivastava, "Dynamic server placement in edge computing toward internet of vehicles," *Computer Communications*, vol. 178, pp. 114–123, 2021.

[38] B. Cao, S. Fan, J. Zhao, S. Tian, Z. Zheng, Y. Yan, and P. Yang, "Large-scale many-objective deployment optimization of edge servers," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3841–3849, 2021.

[39] V. Farhadi, F. Mehmeti, T. He, T. F. La Porta, H. Khamfroush, S. Wang, K. S. Chan, and K. Poularakis, "Service placement and request scheduling for data-intensive applications in edge clouds," *IEEE/ACM Transactions on Networking*, vol. 29, no. 2, pp. 779–792, 2021.

[40] B. Li, P. Hou, H. Wu, and F. Hou, "Optimal edge server deployment and allocation strategy in 5g ultra-dense networking environments," *Pervasive and Mobile Computing*, vol. 72, p. 101312, 2021.

[41] I. Hadžić, Y. Abe, and H. C. Woithe, "Server placement and selection for edge computing in the epc," *IEEE Transactions on Services Computing*, vol. 12, no. 5, pp. 671–684, 2018.

[42] H. Zhu and C. Huang, "Availability-aware mobile edge application placement in 5g networks," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE, 2017, pp. 1–6.

[43] B. Ahat, A. C. Baktır, N. Aras, İ. K. Altınel, A. Özgövde, and C. Ersoy, "Optimal server and service deployment for multi-tier edge cloud computing," *Computer Networks*, vol. 199, p. 108393, 2021.

[44] Z. Amiri, A. Heidari, N. J. Navimipour, and M. Unal, "Resilient and dependability management in distributed environments: A systematic and comprehensive literature review," *Cluster Computing*, vol. 26, no. 2, pp. 1565–1600, 2023.

[45] C. Berger, P. Eichhammer, H. P. Reiser, J. Domaschka, F. J. Hauck, and G. Habiger, "A survey on resilience in the iot: Taxonomy, classification, and discussion of resilience mechanisms," *ACM Computing Surveys (CSUR)*, vol. 54, no. 7, pp. 1–39, 2021.

[46] A. C. Castillo, "An overview resilience in computer networks and network topologies using different metrics," in *Future of Information and Communication Conference*. Springer, 2023, pp. 588–605.

[47] T. Welsh and E. Benkhelifa, "On resilience in cloud computing: A survey of techniques across the cloud domain," *ACM Computing Surveys (CSUR)*, vol. 53, no. 3, pp. 1–36, 2020.

[48] D. S. Fowler, G. Epiphaniou, M. D. Higgins, and C. Maple, "Aspects of resilience for smart manufacturing systems," *Strategic Change*, 2023.

[49] J. Sahoo, M. A. Salahuddin, R. Glitho, H. Elbiaze, and W. Ajib, "A survey on replica server placement algorithms for content delivery networks," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 1002–1026, 2016.

[50] G. Jung, K. Joshi, and S. Ha, "Virtual redundancy for active-standby cloud applications," Sep. 17 2019, uS Patent 10,417,035.

[51] H. H. Esmat, B. Lorenzo, and W. Shi, "Towards resilient network slicing for satellite-terrestrial edge computing iot," *IEEE Internet of Things Journal*, 2023.

[52] P. K. Thiruvasagam, A. Chakraborty, and C. S. R. Murthy, "Resilient and latency-aware orchestration of network slices using multi-connectivity in mec-enabled 5g networks," *IEEE Transactions on Network and Service Management*, vol. 18, no. 3, pp. 2502–2514, 2021.

[53] D. M. Manias, A. Chouman, J. Naoum-Sawaya, and A. Shami, "Resilient and robust qos-preserving post-fault vnf placement," *IEEE Networking Letters*, 2023.

[54] N. Hyodo, T. Sato, R. Shinkuma, and E. Oki, "Resilient virtual network function placement model based on recovery time objectives," in *2020 IEEE 21st International Conference on High Performance Switching and Routing (HPSR)*. IEEE, 2020, pp. 1–7.

[55] J. Xing, J. Gong, X. Foukas, A. Kalia, D. Kim, and M. Kotaru, "Enabling resilience in virtualized rans with atlas," in *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, 2023, pp. 1–15.

[56] Z. H. Nasralla, T. E. Elgorashi, A. Hammadi, M. O. Musa, and J. M. Elmirghani, "Blackout resilient optical core network," *IEEE/ACM Transactions on Networking*, vol. 30, no. 4, pp. 1795–1806, 2022.

[57] Z. Kaleem, M. Yousaf, A. Qamar, A. Ahmad, T. Q. Duong, W. Choi, and A. Jamalipour, "Uav-empowered disaster-resilient edge architecture for delay-sensitive communication," *IEEE Network*, vol. 33, no. 6, pp. 124–132, 2019.

[58] M. Azab, M. Samir, and E. Samir, ""mystify": A proactive moving-target defense for a resilient sdn controller in software defined cps," *Computer Communications*, vol. 189, pp. 205–220, 2022.

[59] M. Aibin, M. Kantor, P. Boryło, H. Niedermayer, P. Chołda, and T. Braun, "Resilient sdn, cdn and icn technology and solutions," *Guide to Disaster-resilient Communication Networks*, pp. 631–652, 2020.

[60] J. Cheng, D. T. Nguyen, and V. K. Bhargava, "Resilient edge service placement under demand and node failure uncertainties," *IEEE Transactions on Network and Service Management*, 2023.

[61] A. Talpur and M. Gurusamy, "Optimizing vehicle-to-edge mapping with load balancing for attack-resilience in iov," in *2023 IEEE 20th Consumer Communications & Networking Conference (CCNC)*. IEEE, 2023, pp. 341–347.

[62] ——, "On attack-resilient service placement and availability in edge-enabled iov networks," *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[63] M. Siew, S. Sharma, and C. Joe-Wong, "Acre: Actor critic reinforcement learning for failure-aware edge computing migrations," in *2023 57th Annual Conference on Information Sciences and Systems (CISS)*. IEEE, 2023, pp. 1–6.

[64] M. Siew, S. Sharma, K. Guo, C. Xu, T. Q. Quek, and C. Joe-Wong, "Fire: A failure-adaptive reinforcement learning framework for edge computing migrations," *arXiv preprint arXiv:2209.14399*, 2022.

[65] J. Moura and D. Hutchison, "Resilience enhancement at edge cloud systems," *IEEE Access*, vol. 10, pp. 45 190–45 206, 2022.

[66] R. Pietrantuono, M. Ficco, and F. Palmieri, "Testing the resilience of mec-based iot applications against resource exhaustion attacks," *IEEE Transactions on Dependable and Secure Computing*, 2023.

[67] M. Masoumi, I. De Miguel, R. J. D. Barroso, L. Ruiz, F. Brasca, G. Rizzi, N. Merayo, J. C. Aguado, P. Fernández, R. M. Lorenzo *et al.*, "Dynamic online vnf placement with different protection schemes in a mec environment," in *2022 32nd International Telecommunication Networks and Applications Conference (ITNAC)*. IEEE, 2022, pp. 1–6.

[68] I. S. M. Isa, T. E. El-Gorashi, M. O. Musa, and J. Elmirghani, "Resilient energy efficient healthcare monitoring infrastructure with server and network protection," *arXiv preprint arXiv:2010.15683*, 2020.

[69] J. Moura and D. Hutchison, "Fog computing systems: State of the art, research issues and future trends, with a focus on resilience," *Journal of Network and Computer Applications*, vol. 169, p. 102784, 2020.

[70] G. Cui, Q. He, X. Xia, F. Chen, H. Jin, and Y. Yang, "Robustness-oriented k edge server placement," in *2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID)*. IEEE, 2020, pp. 81–90.

[71] R. Ford, A. Sridharan, R. Margolies, R. Jana, and S. Rangan, "Provisioning low latency, resilient mobile edge clouds for 5g," in *2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2017, pp. 169–174.

[72] S. N. Shirazi, A. Gouglidis, A. Farshad, and D. Hutchison, "The extended cloud: Review and analysis of mobile edge computing and fog from a security and resilience perspective," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2586–2595, 2017.

[73] C. Colman-Meixner, C. Develder, M. Tornatore, and B. Mukherjee, "A survey on resiliency techniques in cloud computing infrastructures and applications," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 2244–2281, 2016.

[74] M. Dayarathna, Y. Wen, and R. Fan, "Data center energy consumption modeling: A survey," *IEEE Communications surveys & tutorials*, vol. 18, no. 1, pp. 732–794, 2015.

[75] Y. Li, A. Zhou, X. Ma, and S. Wang, "Profit-aware edge server placement," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 55–67, 2021.

[76] X. Zhang, Z. Li, C. Lai, and J. Zhang, "Joint edge server placement and service placement in mobile-edge computing," *IEEE Internet of Things Journal*, vol. 9, no. 13, pp. 11 261–11 274, 2021.

[77] Y. Li and S. Wang, "An energy-aware edge server placement algorithm in mobile edge computing," in *2018 IEEE International conference on edge computing (EDGE)*. IEEE, 2018, pp. 66–73.

[78] Q. Fan and N. Ansari, "Cost aware cloudlet placement for big data processing at the edge," in *2017 IEEE International Conference on Communications (ICC)*. IEEE, 2017, pp. 1–6.

[79] S. Yang, F. Li, M. Shen, X. Chen, X. Fu, and Y. Wang, "Cloudlet placement and task allocation in mobile edge computing," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5853–5863, 2019.

[80] F. Wang, X. Huang, H. Nian, Q. He, Y. Yang, and C. Zhang, "Cost-effective edge server placement in edge computing," in *Proceedings of the 2019 5th international conference on systems, control and Communications*, 2019, pp. 6–10.

[81] L. Su, N. Wang, R. Zhou, and Z. Li, "Dynamic service placement and request scheduling for edge networks," *Computer Networks*, vol. 213, p. 108997, 2022.

[82] Y. Ren, F. Zeng, W. Li, and L. Meng, "A low-cost edge server placement strategy in wireless metropolitan area networks," in *2018 27Th International conference on computer communication and networks (ICCCN)*. IEEE, 2018, pp. 1–6.

[83] F. Zeng, Y. Ren, X. Deng, and W. Li, "Cost-effective edge server placement in wireless metropolitan area networks," *Sensors*, vol. 19, no. 1, p. 32, 2018.

[84] R. Z. Farahani, M. SteadieSeifi, and N. Asgari, "Multiple criteria facility location problems: A survey," *Applied mathematical modelling*, vol. 34, no. 7, pp. 1689–1709, 2010.

[85] L. V. Snyder, "Facility location under uncertainty: a review," *IIE transactions*, vol. 38, no. 7, pp. 547–564, 2006.

[86] J. Deng, J. Guo, and Y. Wang, "A novel k-medoids clustering recommendation algorithm based on probability distribution for collaborative filtering," *Knowledge-Based Systems*, vol. 175, pp. 96–106, 2019.

[87] T. S. Madhulatha, "Comparison between k-means and k-medoids clustering algorithms," in *International Conference on Advances in Computing and Information Technology*. Springer, 2011, pp. 472–481.

[88] N. K. Kaur, U. Kaur, and D. Singh, "K-medoid clustering algorithm-a review," *Int. J. Comput. Appl. Technol*, vol. 1, no. 1, pp. 42–45, 2014.

[89] J. Wang, K. Liu, M. Ni, and J. Pan, "Learning based mobility management under uncertainties for mobile edge computing," in *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2018, pp. 1–6.

[90] G. P. McCormick, "Computability of global solutions to factorable nonconvex programs: Part i—convex underestimating problems," *Mathematical programming*, vol. 10, no. 1, pp. 147–175, 1976.

[91] H. Liu, S. Wang, H. Huang, and Q. Ye, "On the placement of edge servers in mobile edge computing," in *2023 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2023, pp. 496–500.

[92] "Shanghai Telecom Dataset," http://sguangwang.com/TelecomDataset.html, accessed: 2024-10-21.

[93] "Gurobi Optimizer," https://www.gurobi.com/, accessed: 2024-10-21.

[94] K. Wagstaff, C. Cardie, S. Rogers, S. Schrödl *et al.*, "Constrained k-means clustering with background knowledge," in *Icml*, vol. 1, 2001, pp. 577–584.

[95] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 14. Oakland, CA, USA, 1967, pp. 281–297.

Appendix A

Additional Information

A.1 Performance Comparison Using Haversine Distance Metric

In chapter 4, we assess the RESP (heuristic) algorithm's performance by comparing it with established placement strategies. We analyze how it trends relative to the benchmarks (Modified K-means, MIP, Top-K, Shortest-K, Random) using the *Euclidean* distance metric. The following section compares our approach with MIQP [1] using a small dataset. This study is closely related to ours. We provide insights into their performance in similar contexts, using the *Haversine* distance metric. Like our methodology, they also utilized the same dataset and compared their results with other known placement approaches. However, unlike our study, they did not consider resilience in their approach. Based on our considerations, we opted to utilize the *Haversine* distance metric to define the access delay between BSs and ESs for comparing RESP (heuristic) with MIQP. The *Haversine* distance formula accurately calculates distances on the curved surface of the Earth, considering its spherical shape. This formula provides more precise distance calculations, especially over long distances. We adopted kilometers as the unit of the Earth's radius in our calculations. While the *Euclidean* distance is suitable for short distances on flat surfaces, the *Haversine* distance is specifically designed for measuring distances on the surface of a sphere, such as the Earth, offering superior accuracy for longer distances. Our experiments revealed a consistent pattern in the results, regardless of the distance calculation method employed. Specifically, our algorithm consistently followed the same trend irrespective of the chosen distance calculation metric. Furthermore, we demonstrated that our approach consistently outperforms existing methodologies, regardless of the distance calculation metric used.

In our comparison with MIQP, we assessed our approach across two dimensions: (1) utilizing varying numbers of BSs, and (2) considering different numbers of ESs.

A.1.1 Comparison of RESP (heuristic) and MIQP Performance:

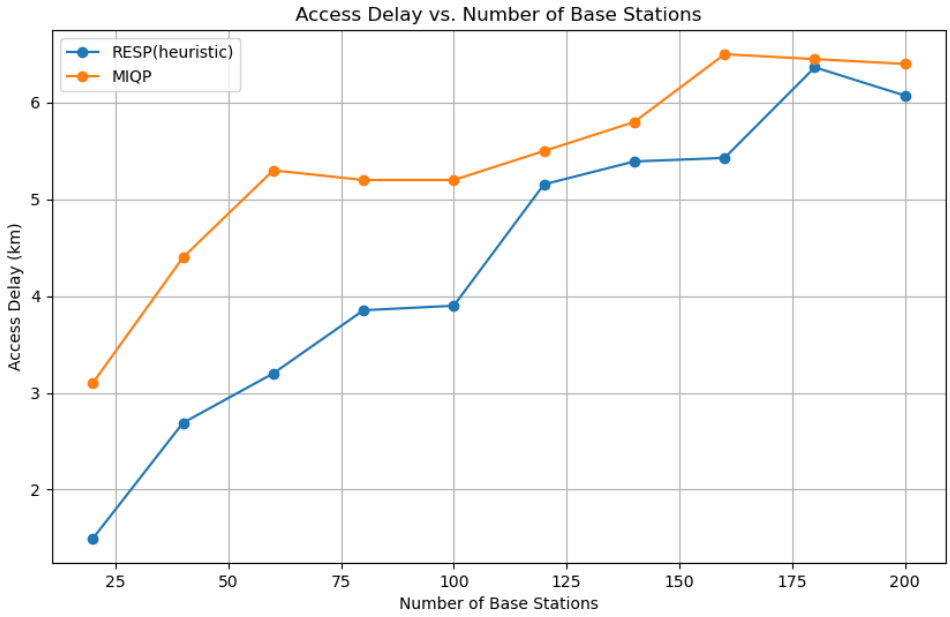
To assess the performance of RESP (heuristic) and MIQP concerning access delay and workload balance, two experiments were devised:

1. In the first experiment, the number of BSs (n) ranged from 20 to 200, while the number of ESs (K) remained constant at 10.

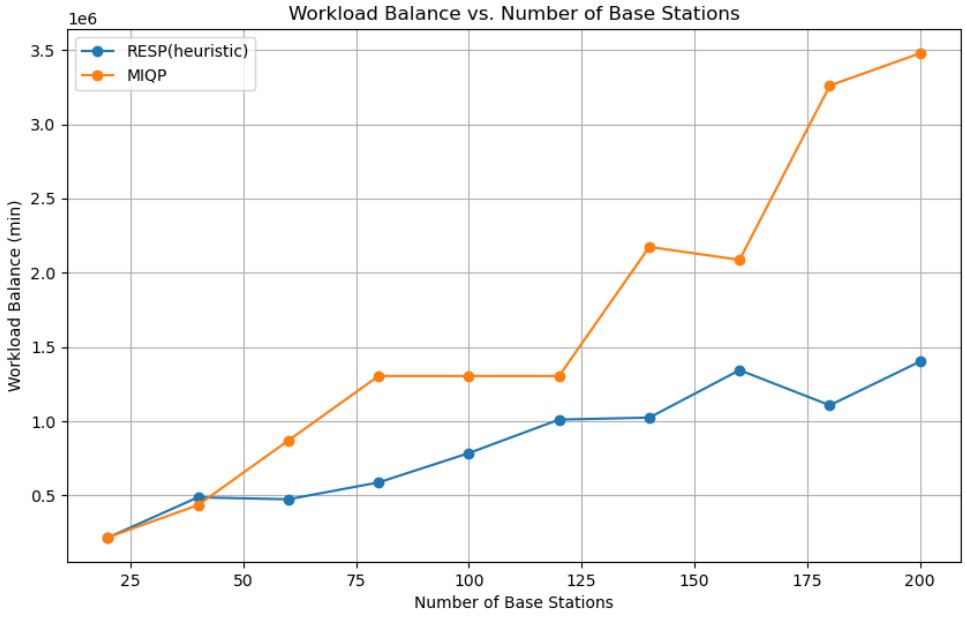
2. In the second experiment, the number of ESs (K) varied from 5 to 30, with the number of BSs (n) fixed at 200.

Comparison of Results Using a Number of BSs: Figure A.1a illustrates the relationship between the number of BSs and the access delay for two different approaches: RESP (heuristic) and MIQP. As depicted in the figure, the access delay tends to increase as the number of BSs rises for both RESP (heuristic) and MIQP. However, RESP (heuristic) generally maintains a lower access delay compared to MIQP across different numbers of BSs. The overall improvement is approximately 21.74%. These findings indicate that RESP (heuristic) consistently outperforms MIQP in minimizing access delay, particularly as the number of BSs increases.

Figure A.1b illustrates the comparative performance of RESP (heuristic) and MIQP in achieving workload balance across various scenarios characterized by different numbers of BSs. Workload balance is measured in minutes, where lower values indicate better workload balance. From the data, we observe that, in scenarios with fewer BSs (e.g., 20 and 40), MIQP generally achieves a better workload balance compared to RESP (heuristic). However, as the number of BSs increases, RESP (heuristic) tends to perform better in workload balance compared to MIQP. This trend continues as the number of BSs further increases. Therefore, the overall improvement in workload balance achieved by RESP (heuristic) compared to MIQP across all scenarios is approximately 36.57%. Continuing our comparison between RESP (heuristic) and MIQP, we now shift our focus from examining the impact of varying BSs to analyzing the effect of different ES configurations. In the following subsection, we explore how changes in the number of ESs influence access delay, providing further insights into the performance differences between the two approaches.



(a) ES access delay

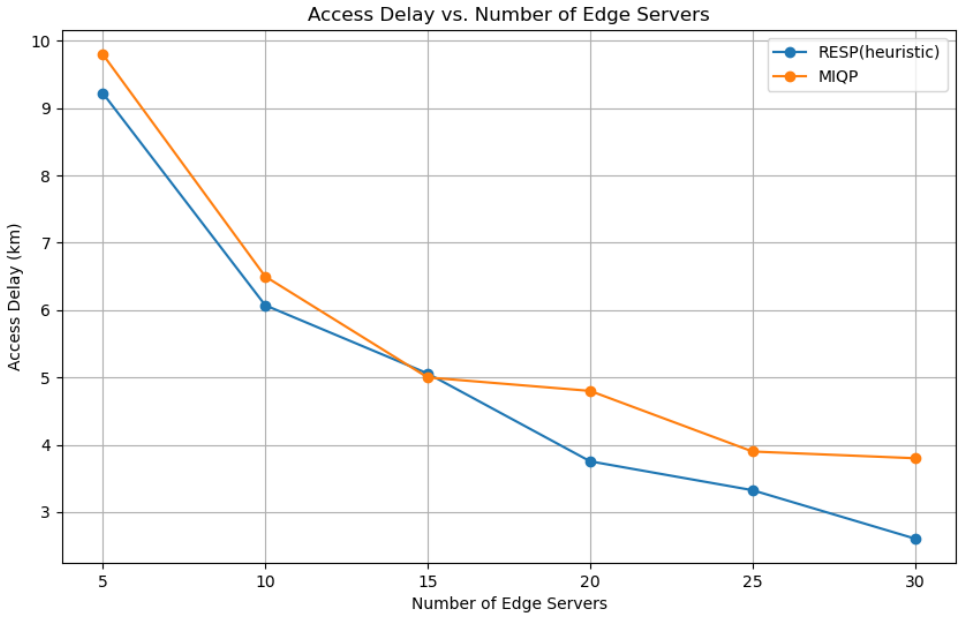


(b) ES workload balancing

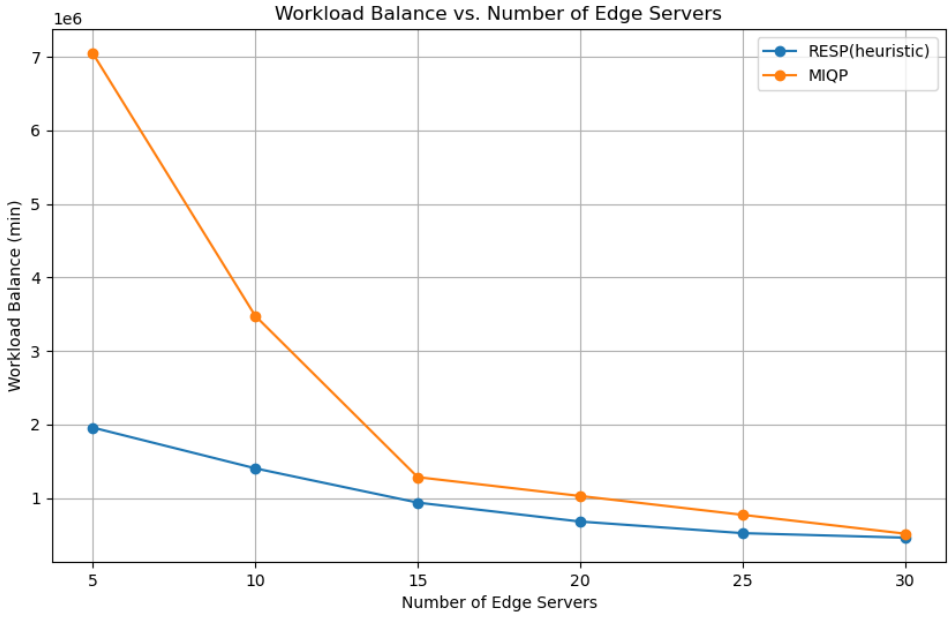
Figure A.1: Performance evaluation of RESP (heuristic) and MIQP with a fixed number of ESs (K = 10) as the number of BSs increases. (a) Access Delay vs. Number of BSs; (b) Workload Balance vs. Number of BSs.

Comparison of Results Using a Number of ESs: Figure A.2a presents a comparison between RESP (heuristic) and MIQP in terms of access delay, considering varying numbers of ESs. As the number of BSs remains constant at 200, the access delay (measured in kilometers) for both RESP (heuristic) and MIQP decreases as the number of ESs increases from 5 to 30. Overall, RESP (heuristic) consistently outperforms MIQP in minimizing access delay across all configurations, demonstrating lower access delay values for each corresponding number of BSs and ESs. Therefore, the overall improvement in access delay achieved by RESP (heuristic) compared to MIQP across all configurations is approximately 13.21%. This trend indicates the superior efficiency of RESP (heuristic) in reducing access delay compared to MIQP, particularly as the number of ESs increases.

Figure A.2b compares the workload balance achieved by RESP (heuristic) and MIQP across different scenarios with a fixed number of BSs (200) and varying numbers of ESs. The workload balance, measured in minutes, is a crucial metric indicating the equitable distribution of computational tasks among ESs. As the number of ESs increases from 5 to 30, both RESP (heuristic) and MIQP demonstrate a reduction in workload imbalance. However, RESP (heuristic) consistently outperforms MIQP in achieving better workload balance across all configurations, with lower values indicating a more even distribution of computational tasks among ESs. The overall improvement in workload balance achieved by RESP (heuristic) compared to MIQP across all configurations is approximately 39.31%. Therefore, the figure highlights the superior effectiveness of RESP (heuristic) in achieving workload balance compared to MIQP, particularly as the number of ESs increases.



(a) ES access delay



(b) ES workload balancing

Figure A.2: Impact of the number of ESs on the performance of RESP (heuristic) and MIQP. (a) Access Delay vs. Number of ESs; (b) Workload Balance vs. Number of ESs.

Table A.1 summarizes the overall percentage of improvement in access delay and workload balance achieved by RESP (heuristic) compared to the MIQP placement strategy across all scenarios.

Table A.1: RESP (heuristic) overall % improvement compared to MIQP.

Serial No.	**Performance Criteria**	**% Improvement Compared to MIQP**
1	Access Delay vs. Number of BSs with a fixed number of ESs (K = 10)	21.74
2	Workload Balance vs. Number of BSs with a fixed number of ESs (K = 10)	36.57
3	Access Delay vs. Number of ESs with a fixed number of BSs (n = 200)	13.21
4	Workload Balance vs. Number of ESs with a fixed number of BSs (n = 200)	39.31

In conclusion, the performance comparison between RESP (heuristic) and MIQP highlights the efficacy of RESP (heuristic) in minimizing access delay and achieving superior workload balance across various network configurations. RESP (heuristic) consistently outperforms MIQP, demonstrating its robustness and efficiency in addressing key optimization objectives within MEC networks. These findings emphasize the potential of RESP (heuristic) as a promising solution for enhancing the performance and resilience of ESP in real-world scenarios.